

付 2. その他の分散並列型用に変更したサブルーチン

付 2.1 データ入力用サブルーチン

ここでは、並列用に変更したサブルーチンで、データ入力に関連するサブルーチンを掲載する。ただし、本文で示したものは除くものとする。

1. Fiber_input_pa

```

C
C
C      ● SUBROUTINE /Fiber_input_pa
C
C
C      ● ファイバー要素の入力(ok)
C
C      ファイバー履歴タイプ
C      nm_type: 1   バイリニア型
C                2   等方硬化+移動硬化トリリニア型
C                3   直線コンクリート型
C                4   曲線コンクリート型
C                5   等方硬化+移動硬化バイリニア型
C                6   非対称トリリニア型
C
C      アナロジーモデル履歴タイプ
C      11  完全弾塑性型
C      12  等方硬化弾塑性型
C      13  移動硬化弾塑性型
C      14  等方硬化+移動硬化弾塑性型
C
C      マルチスプリング履歴タイプ
C      21  武田モデル
C      22  等方硬化+移動硬化トリリニア型
C
C
C      subroutine Fiber_input_pa(it, ierr, n_member, n_element, Member,
*      Element, Model_type, E_model_fiber, M_model_fiber,
*      E_model11, M_model11, E_model12, M_model12,
*      E_model13, M_model13, E_model15, M_model15,
*      E_model21, M_model21, E_model22, M_model22,
*      E_model31, M_model31, E_model32, M_model32, E_model33, M_model33,
*      id_buf, jd_buf, buf, ibuf )
implicit real*8(A-H, O-Z)
include "..%.%sf3st%submain.h"
include "..%.%sf3st%submainx.h"
record / member_s      / Member
record / element_s     / Element
record / E_model11_s   / E_model11
record / E_model12_s   / E_model12
record / E_model13_s   / E_model13
record / E_model15_s   / E_model15
record / E_model21_s   / E_model21

```

```

record / E_model22_s   / E_model22
record / E_model31_s   / E_model31
record / E_model32_s   / E_model32
record / E_model33_s   / E_model33
record / M_model11_s   / M_model11
record / M_model12_s   / M_model12
record / M_model13_s   / M_model13
record / M_model15_s   / M_model15
record / M_model21_s   / M_model21
record / M_model22_s   / M_model22
record / M_model31_s   / M_model31
record / M_model32_s   / M_model32
record / M_model33_s   / M_model33
record / E_model_fiber_s / E_model_fiber
record / M_model_fiber_s / M_model_fiber
record / n_model_s     / Model_type
C
C      ● n_model_s 構造体
C
C      モデルパラメータ
C      structure / n_model_s/
C      integer   n_e_models      ! 要素モデルの最大数
C      integer   no_e_model(100) ! 要素モデルの番号
C      integer   n_div_model(100) ! 要素モデルの分割数
C      integer   n_e_model(100)  ! 要素モデルの数
C      integer   n_m_model(100)  ! 部材モデルの数
C      integer   n_damp(100)     ! 部材減衰
C      integer   n_m_damp        ! 全部材減衰数
C      integer   nm_div_fmodel   ! ファイバー要素の最大数
C      integer   nm_div_felement ! ファイバー要素のエレメント最大数
C      integer   n_m_bilinear    ! ファイバー要素バイリニア用の最大要素数
C      integer   n_m_trilinear   ! ファイバー要素トリリニア用の最大要素数
C      end structure
C      record / n_model_s / Model_type
C      dimension E_model_fiber(*), M_model_fiber(*)
C      dimension E_model11(*), M_model11(*), E_model12(*), M_model12(*)
C      dimension E_model13(*), M_model13(*), E_model15(*), M_model15(*)
C      dimension E_model21(*), M_model21(*), E_model22(*), M_model22(*)
C      dimension E_model31(*), M_model31(*), E_model32(*), M_model32(*)
C      dimension E_model33(*), M_model33(*)
C      dimension Member(*), Element(*)
C      dimension ddm(20)
C      dimension buf(*), ibuf(*)
C      ierr=0
C      if(it.eq.0) then
C
C      read(5,*,err=999) nm      ! 最大断面数
C      write(76,'(a,i4)') ' ファイバー断面総数:', nm
C      ii=0
C
C      ★

```

```

jd_buf=1          ! 並列用
id_buf=0          ! 並列用
c-----★
do i=1,nm
  read(5,*,err=999) n_m,nmm,(ddm(j),j=1,20) ! 断面番号、エレメント数
c  write(76,'(a,i4,a,i4)') '断面番号:',n_m,' ファイバー数:',nmm
  jd_buf=jd_buf + 2 + nmm*2          ! 並列用
  id_buf=id_buf + 20*nmm            ! 並列用
c-----★断面ファイバー数のセット
kk1 = 0
kk2 = 0
kk3 = 0
kk5 = 0
kk21 = 0
kk22 = 0
kk31 = 0
kk32 = 0
kk33 = 0
do i1=1,n_element
  itype_m = Model_type.no_e_model(Element(i1).element_type)
  if(itype_m.eq.11) then
c-----★モデル 1 1
    kk1 = kk1 + 1
    do k=1,2
      if(Element(i1).n_section(k).eq.n_m) then
        Element(i1).nm_section(k) = nmm
        if(k.eq.1) then
          E_model11(kk1).n_section_1 = nmm
          E_model11(kk1).nm_section_1 = ii + 1
        endif
        if(k.eq.2) then
          E_model11(kk1).n_section_2 = nmm
          E_model11(kk1).nm_section_2 = ii+1
        endif
      endif
    enddo
  endif
c-----★モデル 1 2
  if(itype_m.eq.12) then
    kk2 = kk2 + 1
    do k=1,3
      if(Element(i1).n_section(k).eq.n_m) then
        Element(i1).nm_section(k) = nmm
        if(k.eq.1) then
          E_model12(kk2).n_section_1 = nmm
          E_model12(kk2).nm_section_1 = ii + 1
        endif
        if(k.eq.2) then
          E_model12(kk2).n_section_2 = nmm
          E_model12(kk2).nm_section_2 = ii + 1
        endif
      endif
    enddo
  endif

```

```

endif
if(k.eq.3) then
  E_model12(kk2).n_section_c = nmm
  E_model12(kk2).nm_section_c = ii + 1
endif
endif
enddo
endif
c-----★モデル 1 3
if(itype_m.eq.13) then
  kk3 = kk3 + 1
  if(Element(i1).n_section(1).eq.n_m) then
    Element(i1).nm_section(1) = nmm
    E_model13(kk3).n_section_1 = nmm
    E_model13(kk3).nm_section_1 = ii + 1
  endif
endif
c-----★モデル 1 5
if(itype_m.eq.15) then
  kk5 = kk5 + 1
  do k=1,3
    if(Element(i1).n_section(k).eq.n_m) then
      Element(i1).nm_section(k) = nmm
      if(k.eq.1) then
        E_model15(kk5).n_section_1 = nmm
        E_model15(kk5).nm_section_1 = ii + 1
      endif
      if(k.eq.2) then
        E_model15(kk5).n_section_2 = nmm
        E_model15(kk5).nm_section_2 = ii + 1
      endif
    endif
  enddo
endif
c-----★モデル 2 1
if(itype_m.eq.21) then
  kk21 = kk21 + 1
  do k=1,2
    if(Element(i1).n_section(k).eq.n_m) then
      Element(i1).nm_section(k) = nmm
      if(k.eq.1) then
        E_model21(kk21).n_section_1 = nmm
        E_model21(kk21).nm_section_1 = ii + 1
      endif
      if(k.eq.2) then
        E_model21(kk21).n_section_2 = nmm
        E_model21(kk21).nm_section_2 = ii + 1
      endif
    endif
  enddo
endif
endif

```

```

endif
c-----★モデル 2 2
if(itype_m.eq.22) then
kk22=kk22+1
do k=1,3
if(Element(i1).n_section(k).eq.n_m) then
Element(i1).nm_section(k) = nmm
if(k.eq.1) then
E_model22(kk22).n_section_1 = nmm
E_model22(kk22).nm_section_1 = ii + 1
endif
if(k.eq.2) then
E_model22(kk22).n_section_2 = nmm
E_model22(kk22).nm_section_2 = ii + 1
endif
if(k.eq.3) then
E_model22(kk22).n_section_c = nmm
E_model22(kk22).nm_section_c = ii+1
endif
endif
endif
endif
endif
c-----★モデル 3 1
if(itype_m.eq.31) then
kk31 = kk31 + 1
do k=1,2
if(Element(i1).n_section(k).eq.n_m) then
Element(i1).nm_section(k) = nmm
if(k.eq.1) then
E_model31(kk31).n_section_1 = nmm
E_model31(kk31).nm_section_1 = ii + 1
endif
if(k.eq.2) then
E_model31(kk31).n_section_2 = nmm
E_model31(kk31).nm_section_2 = ii + 1
endif
endif
endif
endif
endif
c-----★モデル 3 2
if(itype_m.eq.32) then
kk32=kk32+1
do k=1,3
if(Element(i1).n_section(k).eq.n_m) then
Element(i1).nm_section(k) = nmm
if(k.eq.1) then
E_model32(kk32).n_section_1 = nmm
E_model32(kk32).nm_section_1 = ii + 1
endif
if(k.eq.2) then

```

```

E_model32(kk32).n_section_2 = nmm
E_model32(kk32).nm_section_2 = ii + 1
endif
if(k.eq.3) then
E_model32(kk32).n_section_c = nmm
E_model32(kk32).nm_section_c = ii+1
endif
endif
endif
endif
c-----★モデル 3 3
if(itype_m.eq.33) then
kk33 = kk33 + 1
if(Element(i1).n_section(1).eq.n_m) then
Element(i1).nm_section(1) = nmm
E_model33(kk33).n_section_1 = nmm
E_model33(kk33).nm_section_1 = ii + 1
endif
endif
endif
endif
c-----★
do j=1,nmm
ii = ii + 1
read(5,*,err=999) n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
if(nm_type.le.10) then
goto(901,902,903,904,905,906,907,908,909,910),nm_type
901 continue
goto 900
902 continue
read(5,*,err=999) E_3,Q_2,beta,beta_2 ! 対称トリリニア型 (スチール用)
goto 900
903 continue
read(5,*,err=999) AK_3,AK_4,Q_2,Q_3,Q_4 ! コンクリート型
goto 900
904 continue
read(5,*,err=999) AK_4,Q_3,STR_3,STR_7 ! 曲線コンクリート型
goto 900
905 continue
read(5,*,err=999) beta
goto 900
906 continue
read(5,*,err=999) E_3,Q_2,beta,beta_2
goto 900
907 continue
read(5,*,err=999) Ec_1,Ec_2,Qc_1,beta
goto 900
908 continue
read(5,*,err=999) E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2,beta,beta_2 ! 等方硬化+移動硬化ト
リリニア型
goto 900

```

```

909 continue
    goto 900
910 continue
    goto 900

    elseif(nm_type.le.20) then
        goto(911,912,913,914,915,916,917,918,919,920),nm_type-10
911 continue
    goto 900
912 continue
    goto 900
913 continue
    goto 900
914 continue
    goto 900
915 continue
    goto 900
916 continue
    goto 900
917 continue
    goto 900
918 continue
    goto 900
919 continue
    goto 900
920 continue
    goto 900

    else
        goto(921,922,923,924,925,926,927,928,929,930),nm_type-20
921 continue
    goto 900
922 continue
    goto 900
923 continue
    goto 900
924 continue
    goto 900
925 continue
    goto 900
926 continue
    goto 900
927 continue
    goto 900
928 continue
    goto 900
929 continue
    goto 900
930 continue
    goto 900

```

```

    endif
900 continue
    enddo
    enddo

c-----★要素ファイバー数セット
    Model_type.nm_div_felement= ii
c    write(76,'(a,i5)') ' ファイバー数:',ii
c-----★部材断面ファイバー数のセット
    jj = 0
    do i=1,n_member
        i1 = Member(i).nm_element
        imm = Member(i).n_model_type ! モデルタイプ別番号
        itype_m = Model_type.no_e_model(Element(i1).element_type)
c    write(76,'(a,4i4)') ' fiber ',i,i1,imm,itype_m
c-----★モデル 1 1
        if(itype_m.eq.11) then
            k = 1
            M_model11(imm).n_section_1 = Element(i1).nm_section(k)
            M_model11(imm).nm_section_1 = jj + 1
            jj = jj + Element(i1).nm_section(k)
            k = 2
            M_model11(imm).n_section_2 = Element(i1).nm_section(k)
            M_model11(imm).nm_section_2 = jj + 1
            jj = jj + Element(i1).nm_section(k)
        endif
c-----★モデル 1 2
        if(itype_m.eq.12) then
            k=1
            M_model12(imm).n_section_1 = Element(i1).nm_section(k)
            M_model12(imm).nm_section_1 = jj + 1
            jj = jj + Element(i1).nm_section(k)
            k = 2
            M_model12(imm).n_section_2 = Element(i1).nm_section(k)
            M_model12(imm).nm_section_2 = jj + 1
            jj = jj + Element(i1).nm_section(k)
            k = 3
            M_model12(imm).n_section_c = Element(i1).nm_section(k)
            M_model12(imm).nm_section_c = jj + 1
            jj = jj + Element(i1).nm_section(k)
        endif
c-----★モデル 1 3
        if(itype_m.eq.13) then
            k = 1
            M_model13(imm).n_section_1 = Element(i1).nm_section(k)
            M_model13(imm).nm_section_1 = jj + 1
            jj = jj + Element(i1).nm_section(k)
        endif
c-----★モデル 1 5
        if(itype_m.eq.15) then

```

```

k = 1
M_model15(immm).n_section_1 = Element(i1).nm_section(k)
M_model15(immm).nm_section_1 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 2
M_model15(immm).n_section_2 = Element(i1).nm_section(k)
M_model15(immm).nm_section_2 = jj + 1
jj = jj + Element(i1).nm_section(k)
endif

```

c ★モデル 2 1

```

if(itype_m.eq.21) then
k = 1
M_model21(immm).n_section_1 = Element(i1).nm_section(k)
M_model21(immm).nm_section_1 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 2
M_model21(immm).n_section_2=Element(i1).nm_section(k)
M_model21(immm).nm_section_2=jj+1
jj = jj + Element(i1).nm_section(k)
endif

```

c ★モデル 2 2

```

if(itype_m.eq.22) then
k = 1
M_model22(immm).n_section_1 = Element(i1).nm_section(k)
M_model22(immm).nm_section_1 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 2
M_model22(immm).n_section_2 = Element(i1).nm_section(k)
M_model22(immm).nm_section_2 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 3
M_model22(immm).n_section_c = Element(i1).nm_section(k)
M_model22(immm).nm_section_c = jj + 1
jj = jj + Element(i1).nm_section(k)
endif

```

c ★モデル 3 1

```

if(itype_m.eq.31) then
k = 1
M_model31(immm).n_section_1 = Element(i1).nm_section(k)
M_model31(immm).nm_section_1 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 2
M_model31(immm).n_section_2=Element(i1).nm_section(k)
M_model31(immm).nm_section_2=jj+1
jj = jj + Element(i1).nm_section(k)
endif

```

c ★モデル 3 2

```

if(itype_m.eq.32) then
k = 1
M_model32(immm).n_section_1 = Element(i1).nm_section(k)

```

```

M_model32(immm).nm_section_1 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 2
M_model32(immm).n_section_2 = Element(i1).nm_section(k)
M_model32(immm).nm_section_2 = jj + 1
jj = jj + Element(i1).nm_section(k)
k = 3
M_model32(immm).n_section_c = Element(i1).nm_section(k)
M_model32(immm).nm_section_c = jj + 1
jj = jj + Element(i1).nm_section(k)
endif

```

c ★モデル 3 3

```

if(itype_m.eq.33) then
k = 1
M_model33(immm).n_section_1 = Element(i1).nm_section(k)
M_model33(immm).nm_section_1 = jj + 1
jj = jj + Element(i1).nm_section(k)
endif
enddo
Model_type.nm_div_fmodel = jj      ! ファイバー要素の最大数

```

C

★ ファイバーデータの入力その 2

C

C

c

```

else
read(5,*,err=999) nm
c write(76,'(a,i4)') ' Number of sections:',nm
jd_buf=1      ! 並列用
id_buf=0      ! 並列用
ibuf(jd_buf)=nm      ! 並列用
ii = 0
do i=1,nm
read(5,*,err=999) n_m,nmm,(ddm(j),j=1,20)
jd_buf=jd_buf+2      ! 並列用
ibuf(jd_buf-1)= n_m      ! 並列用
ibuf(jd_buf) = nmm      ! 並列用
c write(76,'(a,2i4,20f10.3)') ' mem:', n_m,nmm,(ddm(j),j=1,20)
do j=1,nmm
read(5,*,err=999) n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az      ! 標準データ
jd_buf=jd_buf+2
ibuf(jd_buf-1)= n      ! 並列用
ibuf(jd_buf) = nm_type      ! 並列用
id_buf=id_buf+9      ! 並列用
buf(id_buf-8)=A      ! 並列用
buf(id_buf-7)=ry      ! 並列用
buf(id_buf-6)=rz      ! 並列用
buf(id_buf-5)=E_1      ! 並列用
buf(id_buf-4)=E_2      ! 並列用
buf(id_buf-3)=Q_1      ! 並列用
buf(id_buf-2)=G      ! 並列用

```

```

      buf(id_buf-1)=Ay      ! 並列用
      buf(id_buf )=Az      ! 並列用
      id_buf=id_buf+11      ! 並列用
c-----★部材断面ファイバー数セット
      ii = ii + 1
      E_model_fiber(ii).nm_type = nm_type      ! 履歴特性番号
      E_model_fiber(ii).E_1 = E_1      ! ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2 = E_2      ! ファイバーの第二剛性 E2
      E_model_fiber(ii).Q_1 = Q_1      ! ファイバーの第一折れ点
      E_model_fiber(ii).G = G      ! ファイバー断面積 G
      E_model_fiber(ii).A = A      ! ファイバー断面積
      E_model_fiber(ii).Ay = Ay      ! ファイバー y 軸せん断断面積
      E_model_fiber(ii).Az = Az      ! ファイバー z 軸せん断断面積
      E_model_fiber(ii).ry = ry      ! 中立軸から断面中心までの y 方向距離
      E_model_fiber(ii).rz = rz      ! 中立軸から断面中心までの z 方向距離

      if(nm_type.le.10) then
        goto(801,802,803,804,805,806,807,808,809,810),nm_type
801 continue
c-----★バイリニア型
c      write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
      E_model_fiber(ii).E_3 = 0.      ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = 0.      ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = E_1      ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0.      ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.      ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0.      ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.      ! ファイバーの圧縮側第二折れ点
      goto 800
802 continue
c-----★トリリニア型
      read(5,*,err=999) E_3,Q_2,beta,beta_2      ! 対称トリリニア型 等方 硬化+移動硬化
トリリニア型
      buf(id_buf-10)=E_3      ! 並列用
      buf(id_buf-9)=Q_2      ! 並列用
      buf(id_buf-8)=beta      ! 並列用
      buf(id_buf-7)=beta_2      ! 並列用
c      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
c      *      E_3,Q_2,beta,beta_2
      E_model_fiber(ii).E_3 = E_3      ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = Q_2      ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = 0.      ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0.      ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.      ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0.      ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.      ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = beta      ! 移動硬化用パラメータ
      E_model_fiber(ii).Beta_2 = beta_2      ! 移動硬化用パラメータその 2
      goto 800
803 continue

```

```

c-----★直線コンクリート型
      read(5,*,err=999) AK_3,AK_4,Q_2,Q_3,Q_4      ! 直線コンクリート型
      buf(id_buf-10)=AK_3      ! 並列用
      buf(id_buf-9)=AK_4      ! 並列用
      buf(id_buf-8)=Q_2      ! 並列用
      buf(id_buf-7)=Q_3      ! 並列用
      buf(id_buf-6)=Q_4      ! 並列用

c      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
c      *      AK_3,AK_4,Q_2,Q_3,Q_4
      E_model_fiber(ii).E_3 = AK_3      ! 圧縮第三勾配
      E_model_fiber(ii).Q_2 = Q_2      ! 圧縮側第一折れ点の応力
      E_model_fiber(ii).Ec_1 = Q_3      ! 圧縮強度
      E_model_fiber(ii).Ec_2 = Q_4      ! 圧縮流れ点
      E_model_fiber(ii).Ec_3 = AK_4      ! 引張第二勾配
      E_model_fiber(ii).Qc_1 = 0.      ! ダミー
      E_model_fiber(ii).Qc_2 = 0.      ! ダミー
      goto 800
804 continue
c-----★曲線コンクリート型
      read(5,*,err=999) AK_4,Q_3,STR_3,STR_7      ! 曲線コンクリート型
      buf(id_buf-10)=AK_4      ! 並列用
      buf(id_buf-9)=Q_3      ! 並列用
      buf(id_buf-8)=STR_3      ! 並列用
      buf(id_buf-7)=STR_7      ! 並列用

c      write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
c      *      AK_4,Q_3,STR_3,STR_7
      E_model_fiber(ii).E_3 = AK_4      ! 引張第二勾配
      E_model_fiber(ii).Q_2 = Q_3      ! 圧縮強度
      E_model_fiber(ii).Ec_1 = STR_3      ! 最大圧縮応力点におけるひずみ量
      E_model_fiber(ii).Ec_2 = STR_7      ! 弾性限界ひずみ量
      E_model_fiber(ii).Ec_3 = 0.      ! ダミー
      E_model_fiber(ii).Qc_1 = 0.      ! ダミー
      E_model_fiber(ii).Qc_2 = 0.      ! ダミー

      goto 800
805 continue
c-----★等方硬化+移動硬化バイリニア型
      read(5,*,err=999) beta      ! 等方硬化+移動硬化バイリニア型
c      write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az      ! 等方硬化+移動硬化バイ
イリニア型
      E_model_fiber(ii).E_3 = 0.      ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = 0.      ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = E_1      ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0.      ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.      ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0.      ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.      ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = beta      ! 移動硬化用パラメータ

```

```

      E_model_fiber(ii).Beta_2 = 0.          ! 移動硬化用パラメータ
      goto 800
806  continue
c-----★等方硬化+移動硬化トリリニア型
      read(5,*,err=999) E_3,Q_2,beta,beta_2 !
      buf(id_buf-10)=E_3          ! 並列用
      buf(id_buf-9)=Q_2           ! 並列用
      buf(id_buf-8)=Ec_1          ! 並列用
      buf(id_buf-7)=Ec_2          ! 並列用
      buf(id_buf-6)=Ec_3          ! 並列用
      buf(id_buf-5)=Qc_1          ! 並列用
      buf(id_buf-4)=Qc_2          ! 並列用
      buf(id_buf-3)=beta          ! 並列用

c  write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
c  *      E_3,Q_2,beta,beta_2
      E_model_fiber(ii).E_3 = E_3          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = Q_2          ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = 0          ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0          ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0          ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0          ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0          ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = beta        ! 移動硬化用パラメータ
      E_model_fiber(ii).Beta_2 = beta_2    ! 移動硬化用パラメータ
      goto 800
807  continue
c-----★非対称バイリニア型
      read(5,*,err=999) Ec_1,Ec_2,Qc_1,beta ! 等方硬化+移動硬化トリリニア型
ア型
c  write(76,'(2i4,15e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
c  *      Ec_1,Ec_2,Qc_1,beta !等方硬化+移動硬化バイリニア型
      buf(id_buf-10)=Ec_1          ! 並列用
      buf(id_buf-9)=Ec_2          ! 並列用
      buf(id_buf-8)=Qc_1          ! 並列用
      buf(id_buf-7)=beta          ! 並列用

      E_model_fiber(ii).E_3 = 0.          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = 0.          ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = Ec_1        ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = Ec_2        ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.          ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = Qc_1        ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.          ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = beta        ! 移動硬化用パラメータ

      goto 800
808  continue
      read(5,*,err=999) E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2,beta,beta_2 ! 等方硬化+移動硬化トリリニア型

```

```

      buf(id_buf-10)=E_3          ! 並列用
      buf(id_buf-9)=Q_2          ! 並列用
      buf(id_buf-8)=Ec_1          ! 並列用
      buf(id_buf-7)=Ec_2          ! 並列用
      buf(id_buf-6)=Ec_3          ! 並列用
      buf(id_buf-5)=Qc_1          ! 並列用
      buf(id_buf-4)=Qc_2          ! 並列用
      buf(id_buf-3)=beta          ! 並列用
      buf(id_buf-2)=beta_2

      E_model_fiber(ii).E_3 = E_3          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = Q_2          ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = Ec_1        ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = Ec_2        ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = Ec_3        ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = Qc_1        ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = Qc_2        ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = beta        ! 移動硬化用パラメータ
      E_model_fiber(ii).Beta_2 = beta_2    ! 移動硬化用パラメータ

      goto 800
809  continue
      goto 800
810  continue
      goto 800

      elseif(nm_type.le.20) then
      goto(811,812,813,814,815,816,817,818,819,820),nm_type-10
811  continue
c  write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
      E_model_fiber(ii).E_1 = A          ! ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2 = ry          ! ファイバーの第二剛性 E2
      E_model_fiber(ii).E_3 = rz          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_1 = E_1        ! ファイバーの第一折れ点
      E_model_fiber(ii).Q_2 = E_2        ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = 0
      E_model_fiber(ii).Ec_2 = 0
      E_model_fiber(ii).Ec_3 = 0
      E_model_fiber(ii).Qc_1 = 0
      E_model_fiber(ii).Qc_2 = 0
      if(ry.eq.0.) E_model_fiber(ii).E_2=A*0.000001
      goto 800
812  continue
c  write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
      E_model_fiber(ii).E_1 = A          ! ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2 = ry          ! ファイバーの第二剛性 E2
      E_model_fiber(ii).E_3 = rz          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_1 = E_1        ! ファイバーの第一折れ点
      E_model_fiber(ii).Q_2 = E_2        ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = 0

```

```

      E_model_fiber(ii).Ec_2 = 0
      E_model_fiber(ii).Ec_3 = 0
      E_model_fiber(ii).Qc_1 = 0
      E_model_fiber(ii).Qc_2 = 0

      if(ry.eq.0.) E_model_fiber(ii).E_2=A*0.000001
      goto 800
813  continue
c    write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
      E_model_fiber(ii).E_1 = A          ! ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2 = ry        ! ファイバーの第二剛性 E2
      E_model_fiber(ii).E_3 = rz        ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_1 = E_1       ! ファイバーの第一折れ点
      E_model_fiber(ii).Q_2 = E_2       ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = 0
      E_model_fiber(ii).Ec_2 = 0
      E_model_fiber(ii).Ec_3 = 0
      E_model_fiber(ii).Qc_1 = 0
      E_model_fiber(ii).Qc_2 = 0

      if(ry.eq.0.) E_model_fiber(ii).E_2=A*0.000001

      goto 800
814  continue
      goto 800
815  continue
      goto 800
816  continue
      goto 800
817  continue
      goto 800
818  continue
      goto 800
819  continue
      goto 800
820  continue
      goto 800
      else
      goto(821,822,823,824,825,826,827,828,829,830),nm_type-20
821  continue
      goto 800
822  continue
      goto 800
823  continue
      goto 800
824  continue
      goto 800
825  continue
      goto 800
826  continue

```

```

      goto 800
827  continue
      goto 800
828  continue
      goto 800
829  continue
      goto 800
830  continue
      goto 800

      endif
800  continue
      enddo

c-----★ファイバーモデル剛性出力
      call Fiber_output(E_model_fiber(ii-nmm+1),nmm)

      enddo

c-----★ファイバー履歴特性数セット
c-----★ファイバー履歴特性数セット

      n_m_bilinear = 0
      n_m_trilinear = 0
      n_m_concrete = 0
      n_m_analogy = 0
      do i=1,n_member
      ie = Member(i).nm_element
      imm = Element(ie).n_element
      im = Member(i).n_model_type

c-----★モデル 1 1
      itype_m = Model_type.no_e_model(Element(ie).element_type)
      if(itype_m.eq.11) then
      ii = E_model11(imm).n_section_1
      nmm = E_model11(imm).nm_section_1 - 1
      nmm=M_model11(im).nm_section_1 - 1
      do j=1,ii
      nmm = nmm + 1
      nmm= nmm + 1
      if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
      n_m_bilinear = n_m_bilinear + 1
      M_model_fiber(nmm).n_type = n_m_bilinear
      elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
      n_m_trilinear = n_m_trilinear + 1
      M_model_fiber(nmm).n_type = n_m_trilinear
      elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
      n_m_concrete = n_m_concrete + 1
      M_model_fiber(nmm).n_type = n_m_concrete

```



```

        endif
    enddo
    ii = E_model11(imm).n_section_2
    nmm = E_model11(imm).nm_section_2 - 1
    nnmm= M_model11(im).nm_section_2 - 1
    do j=1, ii
        nmm = nmm + 1
        nnmm = nnmm + 1
        if(E_model_fiber(nmm).nm_type.eq.1.or.
*         E_model_fiber(nmm).nm_type.eq.5.or.
*         E_model_fiber(nmm).nm_type.eq.7) then
            n_m_bilinear = n_m_bilinear + 1
            M_model_fiber(nnmm).n_type = n_m_bilinear
        elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*         E_model_fiber(nmm).nm_type.eq.6.or.
*         E_model_fiber(nmm).nm_type.eq.8) then
            n_m_trilinear = n_m_trilinear + 1
            M_model_fiber(nnmm).n_type = n_m_trilinear
        elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*         E_model_fiber(nmm).nm_type.eq.4) then
            n_m_concrete = n_m_concrete + 1
            M_model_fiber(nnmm).n_type = n_m_concrete
        endif
    enddo
endif

```

c ————— ★モデル 1 2

```

if(itype_m.eq.12) then
    ii = E_model12(imm).n_section_1
    nmm = E_model12(imm).nm_section_1 - 1
    nnmm= M_model12(im).nm_section_1 - 1
    do j=1, ii
        nmm = nmm + 1
        nnmm = nnmm + 1
        if(E_model_fiber(nmm).nm_type.eq.1.or.
*         E_model_fiber(nmm).nm_type.eq.5.or.
*         E_model_fiber(nmm).nm_type.eq.7) then
            n_m_bilinear = n_m_bilinear + 1
            M_model_fiber(nnmm).n_type = n_m_bilinear
        elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*         E_model_fiber(nmm).nm_type.eq.6.or.
*         E_model_fiber(nmm).nm_type.eq.8) then
            n_m_trilinear = n_m_trilinear + 1
            M_model_fiber(nnmm).n_type = n_m_trilinear
        elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*         E_model_fiber(nmm).nm_type.eq.4) then
            n_m_concrete = n_m_concrete + 1
            M_model_fiber(nnmm).n_type = n_m_concrete
        endif
    enddo
endif
ii = E_model12(imm).n_section_2

```

```

nmm = E_model12(imm).nm_section_2 - 1
nnmm= M_model12(im).nm_section_2 - 1
do j=1, ii
    nmm = nmm + 1
    nnmm= nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.1.or.
*     E_model_fiber(nmm).nm_type.eq.5.or.
*     E_model_fiber(nmm).nm_type.eq.7) then
        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nnmm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*     E_model_fiber(nmm).nm_type.eq.6.or.
*     E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nnmm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*     E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nnmm).n_type = n_m_concrete
    endif
endif

```

```

enddo
ii = E_model12(imm).n_section_c
nmm = E_model12(imm).nm_section_c - 1
nnmm= M_model12(im).nm_section_c - 1
do j=1, ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.1.or.
*     E_model_fiber(nmm).nm_type.eq.5.or.
*     E_model_fiber(nmm).nm_type.eq.7) then
        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nnmm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*     E_model_fiber(nmm).nm_type.eq.6.or.
*     E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nnmm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*     E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nnmm).n_type = n_m_concrete
    endif
endif
enddo
endif

```

c ————— ★モデル 1 3

```

if(itype_m.eq.13) then
    ii = E_model13(imm).n_section_1
    nmm = E_model13(imm).nm_section_1 - 1
    nnmm=M_model13(im).nm_section_1 - 1
    do j=1, ii

```

```

      nmm = nmm + 1
      nnmm = nnmm + 1
      if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nnmm).n_type = n_m_bilinear
      elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nnmm).n_type = n_m_trilinear
      elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nnmm).n_type = n_m_concrete
      endif
    enddo
  endif

```

c-----★モデル 1 5

```

  if(itype_m.eq.15) then
    ii = E_model15(imm).n_section_1
    nmm = E_model15(imm).nm_section_1 - 1
    nnmm = M_model15(im).nm_section_1 - 1
    do j=1,ii
      nmm = nmm + 1
      nnmm = nnmm + 1
      if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nnmm).n_type = n_m_bilinear
      elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nnmm).n_type = n_m_trilinear
      elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nnmm).n_type = n_m_concrete
      endif
    enddo
    ii = E_model15(imm).n_section_2
    nmm = E_model15(imm).nm_section_2 - 1
    nnmm = M_model15(im).nm_section_2 - 1

```

c-----★モデル 2 1

```

  if(itype_m.eq.21) then

```

```

    ii = E_model21(imm).n_section_1
    nmm = E_model21(imm).nm_section_1 - 1
    nnmm = M_model21(im).nm_section_1 - 1
    do j=1,ii
      nmm = nmm + 1
      nnmm = nnmm + 1
      if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nnmm).n_type = n_m_bilinear
      elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nnmm).n_type = n_m_trilinear
      elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nnmm).n_type = n_m_concrete
      endif
    enddo

```

```

    ii = E_model21(imm).n_section_2
    nmm = E_model21(imm).nm_section_2 - 1
    nnmm = M_model21(im).nm_section_2 - 1
    do j=1,ii
      nmm = nmm + 1
      nnmm = nnmm + 1
      if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nnmm).n_type = n_m_bilinear
      elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nnmm).n_type = n_m_trilinear
      elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nnmm).n_type = n_m_concrete
      endif
    enddo
  endif

```

c-----★モデル 2 2

```

  if(itype_m.eq.22) then
    ii = E_model22(imm).n_section_1
    nmm = E_model22(imm).nm_section_1 - 1
    nnmm = M_model22(im).nm_section_1 - 1

```

```

do j=1, ii
  nmm = nmm + 1
  nnmm = nnmm + 1
  if (E_model_fiber(nmm).nm_type.eq.1.or.
*    E_model_fiber(nmm).nm_type.eq.5.or.
*    E_model_fiber(nmm).nm_type.eq.7) then
    n_m_bilinear = n_m_bilinear + 1
    M_model_fiber(nnmm).n_type = n_m_bilinear
  elseif (E_model_fiber(nmm).nm_type.eq.2.or.
*    E_model_fiber(nmm).nm_type.eq.6.or.
*    E_model_fiber(nmm).nm_type.eq.8) then
    n_m_trilinear = n_m_trilinear + 1
    M_model_fiber(nnmm).n_type = n_m_trilinear
  elseif (E_model_fiber(nmm).nm_type.eq.3.or.
*    E_model_fiber(nmm).nm_type.eq.4) then
    n_m_concrete = n_m_concrete + 1
    M_model_fiber(nnmm).n_type = n_m_concrete
  endif
enddo
ii = E_model22(imm).n_section_2
nmm = E_model22(imm).nm_section_2 - 1
nnmm = M_model22(im).nm_section_2 - 1
do j=1, ii
  nmm = nmm + 1
  nnmm = nnmm + 1
  if (E_model_fiber(nmm).nm_type.eq.1.or.
*    E_model_fiber(nmm).nm_type.eq.5.or.
*    E_model_fiber(nmm).nm_type.eq.7) then
    n_m_bilinear = n_m_bilinear + 1
    M_model_fiber(nnmm).n_type = n_m_bilinear
  elseif (E_model_fiber(nmm).nm_type.eq.2.or.
*    E_model_fiber(nmm).nm_type.eq.6.or.
*    E_model_fiber(nmm).nm_type.eq.8) then
    n_m_trilinear = n_m_trilinear + 1
    M_model_fiber(nnmm).n_type = n_m_trilinear
  elseif (E_model_fiber(nmm).nm_type.eq.3.or.
*    E_model_fiber(nmm).nm_type.eq.4) then
    n_m_concrete = n_m_concrete + 1
    M_model_fiber(nnmm).n_type = n_m_concrete
  endif
enddo
ii = E_model22(imm).n_section_c
nmm = E_model22(imm).nm_section_c - 1
nnmm = M_model22(im).nm_section_c - 1
do j=1, ii
  nmm = nmm + 1
  nnmm = nnmm + 1
  if (E_model_fiber(nmm).nm_type.eq.1.or.
*    E_model_fiber(nmm).nm_type.eq.5.or.
*    E_model_fiber(nmm).nm_type.eq.7) then

```

```

    n_m_bilinear = n_m_bilinear + 1
    M_model_fiber(nnmm).n_type = n_m_bilinear
  elseif (E_model_fiber(nmm).nm_type.eq.2.or.
*    E_model_fiber(nmm).nm_type.eq.6.or.
*    E_model_fiber(nmm).nm_type.eq.8) then
    n_m_trilinear = n_m_trilinear + 1
    M_model_fiber(nnmm).n_type = n_m_trilinear
  elseif (E_model_fiber(nmm).nm_type.eq.3.or.
*    E_model_fiber(nmm).nm_type.eq.4) then
    n_m_concrete = n_m_concrete + 1
    M_model_fiber(nnmm).n_type = n_m_concrete
  endif
enddo
endif
c-----★モデル3 1
if (itype_m.eq.31) then
  ii = E_model31(imm).n_section_1
  nmm = E_model31(imm).nm_section_1 - 1
  nnmm = M_model31(im).nm_section_1 - 1
  do j=1, ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if (E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nnmm).n_type = n_m_analogy
    endif
  enddo
  ii = E_model31(imm).n_section_2
  nmm = E_model31(imm).nm_section_2 - 1
  nnmm = M_model31(im).nm_section_2 - 1
  do j=1, ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if (E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nnmm).n_type = n_m_analogy
    endif
  enddo
endif
c-----★モデル3 2
if (itype_m.eq.32) then
  ii = E_model32(imm).n_section_1
  nmm = E_model32(imm).nm_section_1 - 1
  nnmm = M_model32(im).nm_section_1 - 1
  do j=1, ii
    nmm = nmm + 1

```

C _____ ★モデル 33

```

endif
c-----★
enddo
Model_type.n_m_bilinear   = n_m_bilinear
Model_type.n_m_trilinear  = n_m_trilinear
Model_type.n_m_concrete   = n_m_concrete
Model_type.n_m_analogy    = n_m_analogy
write(76,'(a,i8)') '履歴 NO. 1:',n_m_bilinear
write(76,'(a,i8)') '履歴 NO. 2:',n_m_trilinear
write(76,'(a,i8)') '履歴 NO. 3:',n_m_concrete
write(76,'(a,i8)') 'アナロジーモデル:',n_m_analogy
write(76,'(a,i8,a,i8)') 'jd_buf:',jd_buf,'id_buf:',id_buf
endif
return
999 continue
ierr=1
return
end

```

```
C
C
C ● SUBROUTINE /recv_Fiber
C
C
C ● ファイバー要素の入力(ok)
C
C
C   ファイバー履歴タイプ
C   nm_type: 1 バイリニア型
C             2 等方硬化+移動硬化トリリニア型
C             3 直線コンクリート型
C             4 曲線コンクリート型
C             5 等方硬化+移動硬化バイリニア型
C             6 非対称トリリニア型
C
C   アナロジーモデル履歴タイプ
C           11 完全弾塑性型
C           12 等方硬化弾塑性型
C           13 移動硬化弾塑性型
C           14 等方硬化+移動硬化弾塑性型
C
C
C   マルチスプリング履歴タイプ
C           21 武田モデル
C           22 等方硬化+移動硬化トリリニア型
C
```

付-56

```

*      E_model131, M_model131, E_model132, M_model132, E_model133, M_model133,
*      id_buf, jd_buf, buf, ibuf)
c  外部宣言
use MPI_DEFINE
implicit real*8 (A-H, O-Z)
include "...%sf3st%submain.h"
include "...%sf3st%submainx.h"
c  引数宣言
integer      :: it                      ! 制御コード(0:データ受信, 1:データセット)
integer :: ierr, n_member, n_element
record / member_s      / Member
record / element_s     / Element
record / E_model11_s   / E_model11
record / E_model12_s   / E_model12
record / E_model13_s   / E_model13
record / E_model15_s   / E_model15
record / E_model21_s   / E_model21
record / E_model22_s   / E_model22
record / E_model31_s   / E_model31
record / E_model32_s   / E_model32
record / E_model33_s   / E_model33
record / M_model11_s   / M_model11
record / M_model12_s   / M_model12
record / M_model13_s   / M_model13
record / M_model15_s   / M_model15
record / M_model21_s   / M_model21
record / M_model22_s   / M_model22
record / M_model31_s   / M_model31
record / M_model32_s   / M_model32
record / M_model33_s   / M_model33
record / E_model_fiber_s / E_model_fiber
record / M_model_fiber_s / M_model_fiber
record / n_model_s     / Model_type
dimension E_model_fiber(*), M_model_fiber(*)
dimension E_model11(*), M_model11(*), E_model12(*), M_model12(*),
*      E_model15(*), M_model15(*), E_model13(*), M_model13(*)
dimension E_model21(*), M_model21(*), E_model22(*), M_model22(*)
dimension E_model31(*), M_model31(*), E_model32(*), M_model32(*)
dimension E_model33(*), M_model33(*)
dimension Member(*), Element(*)
dimension ddm(20)
real*8 :: buf(*)
integer :: ibuf(*)
c  実装

ierr=0
if(it.eq.0) then
write(76,*) ' MPI_Bcast in 1', jd_buf
write(76,*) ' n_member= ', n_member, ' n_element= ', n_element
c----- ★★ 制御情報と構造用制御情報を受信      ★★-----

```

```

call MPI_Bcast(ibuf, jd_buf, MPI_INTEGER,
+      ID_MASTER, MPI_COMM_WORLD, ierr)
write(76, '(10i4)') ( ibuf(i), i=1, jd_buf)
write(76, '(A, i5)') ' id_buf = ', id_buf
call MPI_Bcast(buf, id_buf, MPI_DOUBLE,
+      ID_MASTER, MPI_COMM_WORLD, ierr)
write(76, '(10f10.2)') ( buf(i), i=1, id_buf)
write(76,*) ' mpi_Bcast out'
c----- ★ファイバーデータの予備入力
ii=0
jd_buf=1                      ! 並列用
id_buf=0                      ! 並列用
nm = ibuf(1)
write(76, '(a, i4)') ' fiber number:', nm
do i=1, nm
jd_buf=jd_buf+2              ! 並列用
nm=ibuf(jd_buf-1)           ! 並列用
nm=ibuf(jd_buf)              ! 並列用
c  read(5,*,err=999) n_m, nmm, (ddm(j), j=1, 20)      ! 断面番号、エレメント数
write(76, '(a, i4, a, i4)') ' 断面番号:', n_m, ' ファイバー数:', nmm
c----- ★断面ファイバー数のセット
kk1 = 0
kk2 = 0
kk5 = 0
kk21 = 0
kk22 = 0
kk31 = 0
kk32 = 0

do i1=1, n_element
itype_m = Model_type.no_e_model(Element(i1).element_type)
if(itype_m.eq.11) then
c----- ★モデル 1 1
kk1 = kk1 + 1
do k=1, 2
if(Element(i1).n_section(k).eq.n_m) then
Element(i1).nm_section(k) = nmm
if(k.eq.1) then
E_model11(kk1).n_section_1 = nmm
E_model11(kk1).nm_section_1 = ii + 1
endif
if(k.eq.2) then
E_model11(kk1).n_section_2 = nmm
E_model11(kk1).nm_section_2 = ii+1
endif
endif
endif
enddo
endif

```

c ★モデル 1 2

```

if(itype_m.eq.12) then
  kk2 = kk2 + 1
  do k=1,3
    if(Element(i1).n_section(k).eq.n_m) then
      Element(i1).nm_section(k) = nmm
      if(k.eq.1) then
        E_model12(kk2).n_section_1 = nmm
        E_model12(kk2).nm_section_1 = ii + 1
      endif
      if(k.eq.2) then
        E_model12(kk2).n_section_2 = nmm
        E_model12(kk2).nm_section_2 = ii + 1
      endif
      if(k.eq.3) then
        E_model12(kk2).n_section_c = nmm
        E_model12(kk2).nm_section_c = ii + 1
      endif
    endif
  enddo
endif

```

c ★モデル 1 3

```

if(itype_m.eq.13) then
  kk3 = kk3 + 1
  if(Element(i1).n_section(1).eq.n_m) then
    Element(i1).nm_section(1) = nmm
    E_model13(kk3).n_section_1 = nmm
    E_model13(kk3).nm_section_1 = ii + 1
  endif
endif

```

c ★モデル 1 5

```

if(itype_m.eq.15) then
  kk5 = kk5 + 1
  do k=1,3
    if(Element(i1).n_section(k).eq.n_m) then
      Element(i1).nm_section(k) = nmm
      if(k.eq.1) then
        E_model15(kk5).n_section_1 = nmm
        E_model15(kk5).nm_section_1 = ii + 1
      endif
      if(k.eq.2) then
        E_model15(kk5).n_section_2 = nmm
        E_model15(kk5).nm_section_2 = ii + 1
      endif
    endif
  enddo
endif

```

c ★モデル 2 1

```

if(itype_m.eq.21) then
  kk21 = kk21 + 1

```

```

do k=1,2
  if(Element(i1).n_section(k).eq.n_m) then
    Element(i1).nm_section(k) = nmm
    if(k.eq.1) then
      E_model21(kk21).n_section_1 = nmm
      E_model21(kk21).nm_section_1 = ii + 1
    endif
    if(k.eq.2) then
      E_model21(kk21).n_section_2 = nmm
      E_model21(kk21).nm_section_2 = ii + 1
    endif
  endif
enddo
endif

```

c ★モデル 2 2

```

if(itype_m.eq.22) then
  kk22=kk22+1
  do k=1,3
    if(Element(i1).n_section(k).eq.n_m) then
      Element(i1).nm_section(k) = nmm
      if(k.eq.1) then
        E_model22(kk22).n_section_1 = nmm
        E_model22(kk22).nm_section_1 = ii + 1
      endif
      if(k.eq.2) then
        E_model22(kk22).n_section_2 = nmm
        E_model22(kk22).nm_section_2 = ii + 1
      endif
      if(k.eq.3) then
        E_model22(kk22).n_section_c = nmm
        E_model22(kk22).nm_section_c = ii+1
      endif
    endif
  enddo
endif

```

c ★モデル 3 1

```

if(itype_m.eq.31) then
  kk31 = kk31 + 1
  do k=1,2
    if(Element(i1).n_section(k).eq.n_m) then
      Element(i1).nm_section(k) = nmm
      if(k.eq.1) then
        E_model31(kk31).n_section_1 = nmm
        E_model31(kk31).nm_section_1 = ii + 1
      endif
      if(k.eq.2) then
        E_model31(kk31).n_section_2 = nmm
        E_model31(kk31).nm_section_2 = ii + 1
      endif
    endif
  enddo
endif

```

```

        enddo
      endif
c-----★モデル 3 2
      if(itype_m.eq.32) then
        kk32=kk32+1
        do k=1,3
          if(Element(i1).n_section(k).eq.n_m) then
            Element(i1).nm_section(k) = nmm
            if(k.eq.1) then
              E_model32(kk32).n_section_1 = nmm
              E_model32(kk32).nm_section_1 = ii + 1
            endif
            if(k.eq.2) then
              E_model32(kk32).n_section_2 = nmm
              E_model32(kk32).nm_section_2 = ii + 1
            endif
            if(k.eq.3) then
              E_model32(kk32).n_section_c = nmm
              E_model32(kk32).nm_section_c = ii+1
            endif
          endif
        enddo
      endif
c-----★モデル 3 3
      if(itype_m.eq.33) then
        kk33 = kk33 + 1
        if(Element(i1).n_section(1).eq.n_m) then
          Element(i1).nm_section(1) = nmm
          E_model33(kk33).n_section_1 = nmm
          E_model33(kk33).nm_section_1 = ii + 1
        endif
      endif
      enddo
c-----★
      do j=1,nmm
        ii = ii + 1
c      read(5,*,err=999) n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
        jd_buf=jd_buf+2
        n= ibuf(jd_buf-1)          ! 並列用
        nm_type= ibuf(jd_buf)      ! 並列用
        id_buf=id_buf+9           ! 並列用
c      buf(id_buf-8)=A            ! 並列用
c      buf(id_buf-7)=ry           ! 並列用
c      buf(id_buf-6)=rz           ! 並列用
c      buf(id_buf-5)=E_1          ! 並列用
c      buf(id_buf-4)=E_2          ! 並列用
c      buf(id_buf-3)=Q_1          ! 並列用
c      buf(id_buf-2)=G            ! 並列用
c      buf(id_buf-1)=Ay           ! 並列用
c      buf(id_buf )=Az            ! 並列用

```

```

        id_buf=id_buf+11          ! 並列用
900 continue
      enddo
      enddo
c-----★要素ファイバー数セット
      Model_type.nm_div_felement= ii
c      write(76,'(a,i5)') ' ファイバー数:',ii
c-----★部材断面ファイバー数のセット
      jj = 0
      do i=1,n_member
        i1 = Member(i).nm_element
        imm = Member(i).n_model_type      ! モデルタイプ別番号
        itype_m = Model_type.no_e_model(Element(i1).element_type)
c-----★モデル 1 1
        if(itype_m.eq.11) then
          k = 1
          M_model11(imm).n_section_1 = Element(i1).nm_section(k)
          M_model11(imm).nm_section_1 = jj + 1
          jj = jj + Element(i1).nm_section(k)
          k = 2
          M_model11(imm).n_section_2 = Element(i1).nm_section(k)
          M_model11(imm).nm_section_2 = jj + 1
          jj = jj + Element(i1).nm_section(k)
        endif
c-----★モデル 1 2
        if(itype_m.eq.12) then
          k=1
          M_model12(imm).n_section_1 = Element(i1).nm_section(k)
          M_model12(imm).nm_section_1 = jj + 1
          jj = jj + Element(i1).nm_section(k)
          k = 2
          M_model12(imm).n_section_2 = Element(i1).nm_section(k)
          M_model12(imm).nm_section_2 = jj + 1
          jj = jj + Element(i1).nm_section(k)
          k = 3
          M_model12(imm).n_section_c = Element(i1).nm_section(k)
          M_model12(imm).nm_section_c = jj + 1
          jj = jj + Element(i1).nm_section(k)
        endif
c-----★モデル 1 3
        if(itype_m.eq.13) then
          k = 1
          M_model13(imm).n_section_1 = Element(i1).nm_section(k)
          M_model13(imm).nm_section_1 = jj + 1
          jj = jj + Element(i1).nm_section(k)
        endif
c-----★モデル 1 5
        if(itype_m.eq.15) then
          k = 1

```

```

      M_model15(immm).n_section_1 = Element(i1).nm_section(k)
      M_model15(immm).nm_section_1 = jj + 1
      jj = jj + Element(i1).nm_section(k)
      k = 2
      M_model15(immm).n_section_2 = Element(i1).nm_section(k)
      M_model15(immm).nm_section_2 = jj + 1
      jj = jj + Element(i1).nm_section(k)
    endif

```

c ★モデル 2 1

```

    if(itype_m.eq.21) then
      k = 1
      M_model21(immm).n_section_1 = Element(i1).nm_section(k)
      M_model21(immm).nm_section_1 = jj + 1
      jj = jj + Element(i1).nm_section(k)
      k = 2
      M_model21(immm).n_section_2=Element(i1).nm_section(k)
      M_model21(immm).nm_section_2=jj+1
      jj = jj + Element(i1).nm_section(k)
    endif

```

c ★モデル 2 2

```

    if(itype_m.eq.22) then
      k = 1
      M_model22(immm).n_section_1 = Element(i1).nm_section(k)
      M_model22(immm).nm_section_1 = jj + 1
      jj = jj + Element(i1).nm_section(k)
      k = 2
      M_model22(immm).n_section_2 = Element(i1).nm_section(k)
      M_model22(immm).nm_section_2 = jj + 1
      jj = jj + Element(i1).nm_section(k)
      k = 3
      M_model22(immm).n_section_c = Element(i1).nm_section(k)
      M_model22(immm).nm_section_c = jj + 1
      jj = jj + Element(i1).nm_section(k)
    endif

```

c ★モデル 3 1

```

    if(itype_m.eq.31) then
      k = 1
      M_model31(immm).n_section_1 = Element(i1).nm_section(k)
      M_model31(immm).nm_section_1 = jj + 1
      jj = jj + Element(i1).nm_section(k)
      k = 2
      M_model31(immm).n_section_2=Element(i1).nm_section(k)
      M_model31(immm).nm_section_2=jj+1
      jj = jj + Element(i1).nm_section(k)
    endif

```

c ★モデル 3 2

```

    if(itype_m.eq.32) then
      k = 1
      M_model32(immm).n_section_1 = Element(i1).nm_section(k)
      M_model32(immm).nm_section_1 = jj + 1

```

```

      jj = jj + Element(i1).nm_section(k)
      k = 2
      M_model32(immm).n_section_2 = Element(i1).nm_section(k)
      M_model32(immm).nm_section_2 = jj + 1
      jj = jj + Element(i1).nm_section(k)
      k = 3
      M_model32(immm).n_section_c = Element(i1).nm_section(k)
      M_model32(immm).nm_section_c = jj + 1
      jj = jj + Element(i1).nm_section(k)
    endif

```

c ★モデル 3 3

```

    if(itype_m.eq.33) then
      k = 1
      M_model33(immm).n_section_1 = Element(i1).nm_section(k)
      M_model33(immm).nm_section_1 = jj + 1
      jj = jj + Element(i1).nm_section(k)
    endif
    enddo
    Model_type.nm_div_fmodel = jj      ! ファイバー要素の最大数

```

C

★ ファイバーデータの入力その 2

C

C

c

```

    else
      c read(5,*,err=999) nm
      c write(76,'(a,i4)') ' Number of sections:',nm
      jd_buf=1 ! 並列用
      id_buf=0 ! 並列用
      nm= ibuf(jd_buf) ! 並列用
      ii = 0
      write(76,'(a,i4)') ' Number of sections:',nm
      do i=1,nm
      c read(5,*,err=999) n_m,nmm,(ddm(j),j=1,20)
      jd_buf=jd_buf+2 ! 並列用
      n_m= ibuf(jd_buf-1) ! 並列用
      nmm= ibuf(jd_buf) ! 並列用
      write(76,'(a,2i4,20f10.3)') ' mem:',n_m,nmm,(ddm(j),j=1,20)
      do j=1,nmm
      c read(5,*,err=999) n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az ! 標準データ
      jd_buf=jd_buf+2
      n= ibuf(jd_buf-1) ! 並列用
      nm_type= ibuf(jd_buf) ! 並列用
      id_buf=id_buf+9 ! 並列用
      A = buf(id_buf-8) ! 並列用
      ry = buf(id_buf-7) ! 並列用
      rz = buf(id_buf-6) ! 並列用
      E_1 = buf(id_buf-5) ! 並列用
      E_2 = buf(id_buf-4) ! 並列用
      Q_1 = buf(id_buf-3) ! 並列用

```



```

      G = buf(id_buf-2)          ! 並列用
      Ay = buf(id_buf-1)        ! 並列用
      Az = buf(id_buf)          ! 並列用
      id_buf=id_buf+11          ! 並列用
c-----★部材断面ファイバー数セット
      ii = ii + 1
      E_model_fiber(ii).nm_type = nm_type          ! 履歴特性番号
      E_model_fiber(ii).E_1 = E_1                  ! ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2 = E_2                  ! ファイバーの第二剛性 E2
      E_model_fiber(ii).Q_1 = Q_1                  ! ファイバーの第一折れ点
      E_model_fiber(ii).G = G                      ! ファイバー断面積 G
      E_model_fiber(ii).A = A                      ! ファイバー断面積
      E_model_fiber(ii).Ay = Ay                    ! ファイバー y 軸せん断用断面積
      E_model_fiber(ii).Az = Az                    ! ファイバー z 軸せん断用断面積
      E_model_fiber(ii).ry = ry                    ! 中立軸から断面中心までの y 方向距離
      E_model_fiber(ii).rz = rz                    ! 中立軸から断面中心までの z 方向距離

      if(nm_type.le.10) then
        goto(801, 802, 803, 804, 805, 806, 807, 808, 809, 810), nm_type
801      continue
c-----★バイリニア型
      write(76, '(2i4, 9e12. 4)') n, nm_type, A, ry, rz, E_1, E_2, Q_1, G, Ay, Az
      E_model_fiber(ii).E_3 = 0.                  ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = 0.                  ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = E_1                ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0.                ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.                ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0.                ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.                ! ファイバーの圧縮側第二折れ点
      goto 800
802      continue
c-----★バイリニア型
c      read(5, *, err=999) E_3, Q_2, beta, beta_2          ! 対称トリリニア型 等方 硬化+移動硬化ト
リリニア型
      E_3 = buf(id_buf-10)          ! 並列用
      Q_2 = buf(id_buf-9)           ! 並列用
      beta = buf(id_buf-8)          ! 並列用
      beta_2 = buf(id_buf-7)        ! 並列用
      write(76, '(2i4, 18e12. 4)') n, nm_type, A, ry, rz, E_1, E_2, Q_1, G, Ay, Az,
      *      E_3, Q_2, beta, beta_2
      E_model_fiber(ii).E_3 = E_3          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = Q_2          ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = 0.          ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0.          ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.          ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0.          ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.          ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = beta        ! 移動硬化用パラメータ
      E_model_fiber(ii).Beta_2 = beta_2    ! 移動硬化用パラメータその 2
      goto 800

```

```

803      continue
c-----★直線コンクリート型
c      read(5, *, err=999) AK_3, AK_4, Q_2, Q_3, Q_4          ! 直線コンクリート型
      AK_3 = buf(id_buf-10)          ! 並列用
      AK_4 = buf(id_buf-9)           ! 並列用
      Q_2 = buf(id_buf-8)            ! 並列用
      Q_3 = buf(id_buf-7)            ! 並列用
      Q_4 = buf(id_buf-6)            ! 並列用

      write(76, '(2i4, 18e12. 4)') n, nm_type, A, ry, rz, E_1, E_2, Q_1, G, Ay, Az,
      *      AK_3, AK_4, Q_2, Q_3, Q_4
      E_model_fiber(ii).E_3 = AK_3          ! 圧縮第三勾配
      E_model_fiber(ii).Q_2 = Q_2           ! 圧縮側第一折れ点の応力
      E_model_fiber(ii).Ec_1 = Q_3          ! 圧縮強度
      E_model_fiber(ii).Ec_2 = Q_4          ! 圧縮流れ点
      E_model_fiber(ii).Ec_3 = AK_4        ! 引張第二勾配
      E_model_fiber(ii).Qc_1 = 0.          ! ダミー
      E_model_fiber(ii).Qc_2 = 0.          ! ダミー
      goto 800
804      continue
c-----★曲線コンクリート型
c      read(5, *, err=999) AK_4, Q_3, STR_3, STR_7          ! 曲線コンクリート型
      AK_4 = buf(id_buf-10)          ! 並列用
      Q_3 = buf(id_buf-9)           ! 並列用
      STR_3 = buf(id_buf-8)          ! 並列用
      STR_7 = buf(id_buf-7)          ! 並列用

      write(76, '(2i4, 18e12. 4)') n, nm_type, A, ry, rz, E_1, E_2, Q_1, G, Ay, Az,
      *      AK_4, Q_3, STR_3, STR_7
      E_model_fiber(ii).E_3 = AK_4          ! 引張第二勾配
      E_model_fiber(ii).Q_2 = Q_3           ! 圧縮強度
      E_model_fiber(ii).Ec_1 = STR_3        ! 最大圧縮応力点におけるひずみ量
      E_model_fiber(ii).Ec_2 = STR_7        ! 弾性限界ひずみ量
      E_model_fiber(ii).Ec_3 = 0.          ! ダミー
      E_model_fiber(ii).Qc_1 = 0.          ! ダミー
      E_model_fiber(ii).Qc_2 = 0.          ! ダミー
      goto 800
805      continue
c-----★等方硬化+移動硬化バイリニア型
c      write(76, '(2i4, 9e12. 4)') n, nm_type, A, ry, rz, E_1, E_2, Q_1, G, Ay, Az          ! 等方硬化+移動硬化バ
イリニア型
      E_model_fiber(ii).E_3 = 0.          ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_2 = 0.          ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1 = E_1        ! ファイバーの圧縮側第一剛性 E1
      E_model_fiber(ii).Ec_2 = 0.          ! ファイバーの圧縮側第二剛性 E2
      E_model_fiber(ii).Ec_3 = 0.          ! ファイバーの圧縮側第三剛性 E3
      E_model_fiber(ii).Qc_1 = 0.          ! ファイバーの圧縮側第一折れ点
      E_model_fiber(ii).Qc_2 = 0.          ! ファイバーの圧縮側第二折れ点
      E_model_fiber(ii).Beta = 0.5        ! 移動硬化用パラメータ

```

```

      goto 800
806  continue
c-----★等方硬化+移動硬化トリリニア型
c  read(5,*,err=999) E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2,beta      !
    E_3 = buf(id_buf-10)      ! 並列用
    Q_2 = buf(id_buf-9)       ! 並列用
    Ec_1 = buf(id_buf-8)      ! 並列用
    Ec_2 = buf(id_buf-7)      ! 並列用
    Ec_3 = buf(id_buf-6)      ! 並列用
    Qc_1 = buf(id_buf-5)      ! 並列用
    Qc_2 = buf(id_buf-4)      ! 並列用
    beta = buf(id_buf-3)      ! 並列用

    write(76,'(2i4,18e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az,
*    E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2
    E_model_fiber(ii).E_3      = E_3      ! ファイバーの第三剛性 E3
    E_model_fiber(ii).Q_2      = Q_2      ! ファイバーの第二折れ点
    E_model_fiber(ii).Ec_1      = Ec_1      ! ファイバーの圧縮側第一剛性 E1
    E_model_fiber(ii).Ec_2      = Ec_2      ! ファイバーの圧縮側第二剛性 E2
    E_model_fiber(ii).Ec_3      = Ec_3      ! ファイバーの圧縮側第三剛性 E3
    E_model_fiber(ii).Qc_1      = Qc_1      ! ファイバーの圧縮側第一折れ点
    E_model_fiber(ii).Qc_2      = Qc_2      ! ファイバーの圧縮側第二折れ点
    E_model_fiber(ii).Beta      = beta      ! 移動硬化用パラメータ
    goto 800
807  continue
c-----★非対称バイリニア型
c  read(5,*,err=999) Ec_1,Ec_2,Qc_1,beta      ! 等方硬化+移動硬化トリリニア型
    Ec_1 = buf(id_buf-10)      ! 並列用
    Ec_2 = buf(id_buf-9)       ! 並列用
    Qc_1 = buf(id_buf-8)       ! 並列用
    beta = buf(id_buf-7)       ! 並列用

    E_model_fiber(ii).E_3      = 0.      ! ファイバーの第三剛性 E3
    E_model_fiber(ii).Q_2      = 0.      ! ファイバーの第二折れ点
    E_model_fiber(ii).Ec_1      = Ec_1      ! ファイバーの圧縮側第一剛性 E1
    E_model_fiber(ii).Ec_2      = Ec_2      ! ファイバーの圧縮側第二剛性 E2
    E_model_fiber(ii).Ec_3      = 0.      ! ファイバーの圧縮側第三剛性 E3
    E_model_fiber(ii).Qc_1      = Qc_1      ! ファイバーの圧縮側第一折れ点
    E_model_fiber(ii).Qc_2      = 0.      ! ファイバーの圧縮側第二折れ点
    E_model_fiber(ii).Beta      = beta      ! 移動硬化用パラメータ

    goto 800
808  continue
    read(5,*,err=999) E_3,Q_2,Ec_1,Ec_2,Ec_3,Qc_1,Qc_2,beta,beta_2 ! 等方硬化+移動硬化トリリニア型
    E_3 = buf(id_buf-10)      ! 並列用
    Q_2 = buf(id_buf-9)       ! 並列用
    Ec_1 = buf(id_buf-8)      ! 並列用
    Ec_2 = buf(id_buf-7)      ! 並列用

```

```

    Ec_3 = buf(id_buf-6)      ! 並列用
    Qc_1 = buf(id_buf-5)      ! 並列用
    Qc_2 = buf(id_buf-4)      ! 並列用
    beta = buf(id_buf-3)      ! 並列用
    beta_2 = buf(id_buf-2)

    E_model_fiber(ii).E_3      = E_3      ! ファイバーの第三剛性 E3
    E_model_fiber(ii).Q_2      = Q_2      ! ファイバーの第二折れ点
    E_model_fiber(ii).Ec_1      = Ec_1      ! ファイバーの圧縮側第一剛性 E1
    E_model_fiber(ii).Ec_2      = Ec_2      ! ファイバーの圧縮側第二剛性 E2
    E_model_fiber(ii).Ec_3      = Ec_3      ! ファイバーの圧縮側第三剛性 E3
    E_model_fiber(ii).Qc_1      = Qc_1      ! ファイバーの圧縮側第一折れ点
    E_model_fiber(ii).Qc_2      = Qc_2      ! ファイバーの圧縮側第二折れ点
    E_model_fiber(ii).Beta      = beta      ! 移動硬化用パラメータ
    E_model_fiber(ii).Beta_2    = beta_2    ! 移動硬化用パラメータ

    goto 800
809  continue
    goto 800
810  continue
    goto 800

    elseif(nm_type.le.20) then
    goto(811,812,813,814,815,816,817,818,819,820),nm_type-10
811  continue
    write(76,'(2i4,9e12.4)') n,nm_type,A,ry,rz,E_1,E_2,Q_1,G,Ay,Az
    E_model_fiber(ii).E_1      = A      ! ファイバーの第一剛性 E1
    E_model_fiber(ii).E_2      = ry      ! ファイバーの第二剛性 E2
    E_model_fiber(ii).E_3      = rz      ! ファイバーの第三剛性 E3
    E_model_fiber(ii).Q_1      = E_1      ! ファイバーの第一折れ点
    E_model_fiber(ii).Q_2      = E_2      ! ファイバーの第二折れ点
    E_model_fiber(ii).Ec_1      = 0
    E_model_fiber(ii).Ec_2      = 0
    E_model_fiber(ii).Ec_3      = 0
    E_model_fiber(ii).Qc_1      = 0
    E_model_fiber(ii).Qc_2      = 0

    if(ry.eq.0.) E_model_fiber(ii).E_2=A*0.000001
    goto 800
812  continue
    E_model_fiber(ii).E_1      = A      ! ファイバーの第一剛性 E1
    E_model_fiber(ii).E_2      = ry      ! ファイバーの第二剛性 E2
    E_model_fiber(ii).E_3      = rz      ! ファイバーの第三剛性 E3
    E_model_fiber(ii).Q_1      = E_1      ! ファイバーの第一折れ点
    E_model_fiber(ii).Q_2      = E_2      ! ファイバーの第二折れ点
    E_model_fiber(ii).Ec_1      = 0
    E_model_fiber(ii).Ec_2      = 0
    E_model_fiber(ii).Ec_3      = 0
    E_model_fiber(ii).Qc_1      = 0
    E_model_fiber(ii).Qc_2      = 0

```

```

      if(ry.eq.0.) E_model_fiber(ii).E_2=A*0.000001
      goto 800
813  continue
      E_model_fiber(ii).E_1      = A          ! ファイバーの第一剛性 E1
      E_model_fiber(ii).E_2      = ry         ! ファイバーの第二剛性 E2
      E_model_fiber(ii).E_3      = rz         ! ファイバーの第三剛性 E3
      E_model_fiber(ii).Q_1      = E_1        ! ファイバーの第一折れ点
      E_model_fiber(ii).Q_2      = E_2        ! ファイバーの第二折れ点
      E_model_fiber(ii).Ec_1     = 0
      E_model_fiber(ii).Ec_2     = 0
      E_model_fiber(ii).Ec_3     = 0
      E_model_fiber(ii).Qc_1     = 0
      E_model_fiber(ii).Qc_2     = 0

      if(ry.eq.0.) E_model_fiber(ii).E_2=A*0.000001

      goto 800
814  continue
      goto 800
815  continue
      goto 800
816  continue
      goto 800
817  continue
      goto 800
818  continue
      goto 800
819  continue
      goto 800
820  continue
      goto 800

      else
      goto(821, 822, 823, 824, 825, 826, 827, 828, 829, 830), nm_type-20
821  continue
      goto 800
822  continue
      goto 800
823  continue
      goto 800
824  continue
      goto 800
825  continue
      goto 800
826  continue
      goto 800
827  continue
      goto 800
828  continue

```

```

      goto 800
829  continue
      goto 800
830  continue
      goto 800

      endif
800  continue
      enddo
      enddo

c-----★ファイバー履歴特性数セット
      n_m_bilinear      = 0
      n_m_trilinear     = 0
      n_m_concrete      = 0
      n_m_analogy       = 0
      do i=1,n_member
      ie = Member(i).nm_element
      imm = Element(ie).n_element
      im = Member(i).n_model_type

c-----★モデル 1 1
      itype_m = Model_type.no_e_model(Element(ie).element_type)
      write(76,'(a,6i4)') 'i,ie,imm,im',i,ie,imm,im,itype_m,
*      Element(ie).element_type
      if(itype_m.eq.11) then
      ii = E_model11(imm).n_section_1
      nmm = E_model11(imm).nm_section_1 - 1
      nrmm=M_model11(im).nm_section_1 - 1
      do j=1,ii
      nmm = nmm + 1
      nrmm= nrmm + 1
      if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
      n_m_bilinear = n_m_bilinear + 1
      M_model_fiber(nrmm).n_type = n_m_bilinear
      elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
      n_m_trilinear = n_m_trilinear + 1
      M_model_fiber(nrmm).n_type = n_m_trilinear
      elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
      n_m_concrete = n_m_concrete + 1
      M_model_fiber(nrmm).n_type = n_m_concrete
      endif
      enddo
      ii = E_model11(imm).n_section_2
      nmm = E_model11(imm).nm_section_2 - 1
      nrmm= M_model11(im).nm_section_2 - 1

```

```

do j=1,ii
  nmm = nmm + 1
  nnmm = nnmm + 1
  if(E_model_fiber(nmm).nm_type.eq.1.or.
*    E_model_fiber(nmm).nm_type.eq.5.or.
*    E_model_fiber(nmm).nm_type.eq.7) then
    n_m_bilinear = n_m_bilinear + 1
    M_model_fiber(nnmm).n_type = n_m_bilinear
  elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*    E_model_fiber(nmm).nm_type.eq.6.or.
*    E_model_fiber(nmm).nm_type.eq.8) then
    n_m_trilinear = n_m_trilinear + 1
    M_model_fiber(nnmm).n_type = n_m_trilinear
  elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*    E_model_fiber(nmm).nm_type.eq.4) then
    n_m_concrete = n_m_concrete + 1
    M_model_fiber(nnmm).n_type = n_m_concrete
  endif
enddo
endif

```

c ★モデル 1 2

```

if(itype_m.eq.12) then
  ii = E_model12(imm).n_section_1
  nmm = E_model12(imm).nm_section_1 - 1
  nnmm= M_model12(im).nm_section_1 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm= nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
      n_m_bilinear = n_m_bilinear + 1
      M_model_fiber(nnmm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
      n_m_trilinear = n_m_trilinear + 1
      M_model_fiber(nnmm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
      n_m_concrete = n_m_concrete + 1
      M_model_fiber(nnmm).n_type = n_m_concrete
    endif
  enddo
  ii = E_model12(imm).n_section_2
  nmm = E_model12(imm).nm_section_2 - 1
  nnmm= M_model12(im).nm_section_2 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm= nnmm + 1

```

```

if(E_model_fiber(nmm).nm_type.eq.1.or.
*   E_model_fiber(nmm).nm_type.eq.5.or.
*   E_model_fiber(nmm).nm_type.eq.7) then
  n_m_bilinear = n_m_bilinear + 1
  M_model_fiber(nnmm).n_type = n_m_bilinear
elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*   E_model_fiber(nmm).nm_type.eq.6.or.
*   E_model_fiber(nmm).nm_type.eq.8) then
  n_m_trilinear = n_m_trilinear + 1
  M_model_fiber(nnmm).n_type = n_m_trilinear
elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*   E_model_fiber(nmm).nm_type.eq.4) then
  n_m_concrete = n_m_concrete + 1
  M_model_fiber(nnmm).n_type = n_m_concrete
endif
enddo
ii = E_model12(imm).n_section_c
nmm = E_model12(imm).nm_section_c - 1
nnmm= M_model12(im).nm_section_c - 1
do j=1,ii
  nmm = nmm + 1
  nnmm = nnmm + 1
  if(E_model_fiber(nmm).nm_type.eq.1.or.

```

```

*   E_model_fiber(nmm).nm_type.eq.5.or.
*   E_model_fiber(nmm).nm_type.eq.7) then
    n_m_bilinear = n_m_bilinear + 1
    M_model_fiber(nnmm).n_type = n_m_bilinear
  elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*   E_model_fiber(nmm).nm_type.eq.6.or.
*   E_model_fiber(nmm).nm_type.eq.8) then
    n_m_trilinear = n_m_trilinear + 1
    M_model_fiber(nnmm).n_type = n_m_trilinear
  elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*   E_model_fiber(nmm).nm_type.eq.4) then
    n_m_concrete = n_m_concrete + 1
    M_model_fiber(nnmm).n_type = n_m_concrete
  endif
enddo
endif

```

c ★モデル 1 3

```

if(itype_m.eq.13) then
  ii = E_model13(imm).n_section_1
  nmm = E_model13(imm).nm_section_1 - 1
  nnmm= M_model13(im).nm_section_1 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm= nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then

```

```

        n_m_bilinear = n_m_bilinear + 1
        M_model_fiber(nmmm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*       E_model_fiber(nmm).nm_type.eq.6.or.
*       E_model_fiber(nmm).nm_type.eq.8) then
        n_m_trilinear = n_m_trilinear + 1
        M_model_fiber(nmmm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*       E_model_fiber(nmm).nm_type.eq.4) then
        n_m_concrete = n_m_concrete + 1
        M_model_fiber(nmmm).n_type = n_m_concrete
    endif
enddo
endif

```

c-----★モデル15

```

if(itype_m.eq.15) then
    ii = E_model15(imm).n_section_1
    nmm = E_model15(imm).nm_section_1 - 1
    nmmm= M_model15(im).nm_section_1 - 1
    do j=1,ii
        nmm = nmm + 1
        nmmm = nmmm + 1
        if(E_model_fiber(nmm).nm_type.eq.1.or.
*       E_model_fiber(nmm).nm_type.eq.5.or.
*       E_model_fiber(nmm).nm_type.eq.7) then
            n_m_bilinear = n_m_bilinear + 1
            M_model_fiber(nmmm).n_type = n_m_bilinear
        elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*       E_model_fiber(nmm).nm_type.eq.6.or.
*       E_model_fiber(nmm).nm_type.eq.8) then
            n_m_trilinear = n_m_trilinear + 1
            M_model_fiber(nmmm).n_type = n_m_trilinear
        elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*       E_model_fiber(nmm).nm_type.eq.4) then
            n_m_concrete = n_m_concrete + 1
            M_model_fiber(nmmm).n_type = n_m_concrete
        endif
    enddo
    ii = E_model15(imm).n_section_2
    nmm = E_model15(imm).nm_section_2 - 1
    nmmm= M_model15(im).nm_section_2 - 1

```

endif

c-----★モデル21

```

if(itype_m.eq.21) then
    ii = E_model21(imm).n_section_1
    nmm = E_model21(imm).nm_section_1 - 1
    nmmm= M_model21(im).nm_section_1 - 1
    do j=1,ii
        nmm = nmm + 1

```

```

        nmmm = nmmm + 1
        if(E_model_fiber(nmm).nm_type.eq.1.or.
*       E_model_fiber(nmm).nm_type.eq.5.or.
*       E_model_fiber(nmm).nm_type.eq.7) then
            n_m_bilinear = n_m_bilinear + 1
            M_model_fiber(nmmm).n_type = n_m_bilinear
        elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*       E_model_fiber(nmm).nm_type.eq.6.or.
*       E_model_fiber(nmm).nm_type.eq.8) then
            n_m_trilinear = n_m_trilinear + 1
            M_model_fiber(nmmm).n_type = n_m_trilinear
        elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*       E_model_fiber(nmm).nm_type.eq.4) then
            n_m_concrete = n_m_concrete + 1
            M_model_fiber(nmmm).n_type = n_m_concrete
        endif
    enddo
    ii = E_model21(imm).n_section_2
    nmm = E_model21(imm).nm_section_2 - 1
    nmmm= M_model21(im).nm_section_2 - 1
    do j=1,ii
        nmm = nmm + 1
        nmmm = nmmm + 1
        if(E_model_fiber(nmm).nm_type.eq.1.or.
*       E_model_fiber(nmm).nm_type.eq.5.or.
*       E_model_fiber(nmm).nm_type.eq.7) then
            n_m_bilinear = n_m_bilinear + 1
            M_model_fiber(nmmm).n_type = n_m_bilinear
        elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*       E_model_fiber(nmm).nm_type.eq.6.or.
*       E_model_fiber(nmm).nm_type.eq.8) then
            n_m_trilinear = n_m_trilinear + 1
            M_model_fiber(nmmm).n_type = n_m_trilinear
        elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*       E_model_fiber(nmm).nm_type.eq.4) then
            n_m_concrete = n_m_concrete + 1
            M_model_fiber(nmmm).n_type = n_m_concrete
        endif
    enddo
endif

```

c-----★モデル22

```

if(itype_m.eq.22) then
    ii = E_model22(imm).n_section_1
    nmm = E_model22(imm).nm_section_1 - 1
    nmmm= M_model22(im).nm_section_1 - 1
    do j=1,ii
        nmm = nmm + 1
        nmmm = nmmm + 1
        if(E_model_fiber(nmm).nm_type.eq.1.or.
*       E_model_fiber(nmm).nm_type.eq.5.or.

```

```

*      E_model_fiber(nmm).nm_type.eq.7) then
      n_m_bilinear = n_m_bilinear + 1
      M_model_fiber(nnm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
      n_m_trilinear = n_m_trilinear + 1
      M_model_fiber(nnm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
      n_m_concrete = n_m_concrete + 1
      M_model_fiber(nnm).n_type = n_m_concrete
    endif
  enddo
  ii = E_model22(imm).n_section_2
  nmm = E_model22(imm).nm_section_2 - 1
  nnmm = M_model22(im).nm_section_2 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
      n_m_bilinear = n_m_bilinear + 1
      M_model_fiber(nnm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then
      n_m_trilinear = n_m_trilinear + 1
      M_model_fiber(nnm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
      n_m_concrete = n_m_concrete + 1
      M_model_fiber(nnm).n_type = n_m_concrete
    endif
  enddo
  ii = E_model22(imm).n_section_c
  nmm = E_model22(imm).nm_section_c - 1
  nnmm = M_model22(im).nm_section_c - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.1.or.
*      E_model_fiber(nmm).nm_type.eq.5.or.
*      E_model_fiber(nmm).nm_type.eq.7) then
      n_m_bilinear = n_m_bilinear + 1
      M_model_fiber(nnm).n_type = n_m_bilinear
    elseif(E_model_fiber(nmm).nm_type.eq.2.or.
*      E_model_fiber(nmm).nm_type.eq.6.or.
*      E_model_fiber(nmm).nm_type.eq.8) then

```

```

      n_m_trilinear = n_m_trilinear + 1
      M_model_fiber(nnm).n_type = n_m_trilinear
    elseif(E_model_fiber(nmm).nm_type.eq.3.or.
*      E_model_fiber(nmm).nm_type.eq.4) then
      n_m_concrete = n_m_concrete + 1
      M_model_fiber(nnm).n_type = n_m_concrete
    endif
  enddo
endif
c-----★モデル 3 1
if(itype_m.eq.31) then
  ii = E_model31(imm).n_section_1
  nmm = E_model31(imm).nm_section_1 - 1
  nnmm = M_model31(im).nm_section_1 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nnm).n_type = n_m_analogy
    endif
  enddo
  ii = E_model31(imm).n_section_2
  nmm = E_model31(imm).nm_section_2 - 1
  nnmm = M_model31(im).nm_section_2 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nnm).n_type = n_m_analogy
    endif
  enddo
endif
c-----★モデル 3 2
if(itype_m.eq.32) then
  ii = E_model32(imm).n_section_1
  nmm = E_model32(imm).nm_section_1 - 1
  nnmm = M_model32(im).nm_section_1 - 1
  do j=1,ii
    nmm = nmm + 1
    nnmm = nnmm + 1
    if(E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1

```

```

      M_model_fiber(nmmm).n_type = n_m_analogy
    endif
  enddo
  ii = E_model32(imm).n_section_2
  nmm = E_model32(imm).nm_section_2 - 1
  nmmm= M_model32(im).nm_section_2 - 1
  do j=1,ii
    nmm = nmm + 1
    nmmm = nmmm + 1
    if(E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nmmm).n_type = n_m_analogy
    endif
  enddo
  ii = E_model32(imm).n_section_c
  nmm = E_model32(imm).nm_section_c - 1
  nmmm= M_model32(im).nm_section_c - 1
  do j=1,ii
    nmm = nmm + 1
    nmmm = nmmm + 1
    if(E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nmmm).n_type = n_m_analogy
    endif
  enddo
endif
c-----★モデル3 3
if(itype_m.eq.33) then
  ii = E_model33(imm).n_section_1
  nmm = E_model33(imm).nm_section_1 - 1
  nmmm=M_model33(im).nm_section_1 - 1
  do j=1,ii
    nmm = nmm + 1
    nmmm= nmmm + 1
    if(E_model_fiber(nmm).nm_type.eq.11.or.
*      E_model_fiber(nmm).nm_type.eq.12.or.
*      E_model_fiber(nmm).nm_type.eq.13) then
      n_m_analogy = n_m_analogy + 1
      M_model_fiber(nmmm).n_type = n_m_analogy
    endif
  enddo
endif
c-----★
enddo
Model_type.n_m_bilinear = n_m_bilinear
Model_type.n_m_trilinear = n_m_trilinear

```

```

Model_type.n_m_concrete = n_m_concrete
Model_type.n_m_analogy = n_m_analogy
write(76,'(a,i8)') ' 履歴 NO.1:',n_m_bilinear
write(76,'(a,i8)') ' 履歴 NO.2:',n_m_trilinear
write(76,'(a,i8)') ' 履歴 NO.3:',n_m_concrete
write(76,'(a,i8)') ' アナロジーモデル:',n_m_analogy
endif
return
999 continue
ierr=1
return
end

```

3. Get_structure_pa

```

C
C ● SUBROUTINE /Get_structure_pa
C
C ● 構造データを入力し、バッファにデータをセットする
C
c-----★計算結果のダンプ出力ファイル番号
subroutine Get_structure_pa(Point,Member,Element,Parameter_C,
*      Model_type,ierr,buf,ibuf,iis,jjs,
*      S_comp_model,E_Fiber_work)
parameter(damp_out = 76)
implicit real*8(A-H,O-Z)
include "..¥..¥sf3st¥submain.h"
include "..¥..¥sf3st¥New_submain.h"
record / E_Fiber_work_s / E_Fiber_work
record / S_comp_model_s / S_comp_model
record / parameter_s / Parameter_C
record / member_s / Member
record / point_s / Point
record / element_s / Element
record / n_model_s / Model_type
dimension n_section(100),E_Fiber_work(*),S_comp_model(*)
dimension Member(*),Point(*),Element(*)
character title*80
dimension ire(6) ! ワークエリアとして使用
integer RO_MODEL_NUMBER,TRI_MODEL_NUMBER

C
integer ibuf(*)
dimension buf(*)

C
data
RO_MODEL_NUMBER/12/
data TRI_MODEL_NUMBER/11/

C
c バッファデータ仕様
c buf() 1 : 座標 3
c 2 : 局所座標 3

```

```

c      3 : 要素 17
c      4 : 部材 4
c      ibuf() 1 : 境界拘束条件 6
c      2 : 要素 6
c      3 : 部材 14
c
c
c
c
c      Parameter_C : 構造体
c      Member      : 構造体
c      Element      : 構造体
c      Point        : 構造体
c      Mode_type    : 構造体
c      構造体の定義は、インクルードファイル「submain.h」で行われている。
c
c      節点構造体
c      structure / point_s/
c      real*8 coord(3) ! 3次元の節点座標
c      integer irest(6) ! 節点自由度の拘束表
c      integer local_coord ! 局所座標系の有無（ある場合は、回転行列の通し番号）
c      real*8 coord_local(3) ! 局所座標系（全体座標系に対する角度）
c      real*8 disp_initial(3) ! 初期変位
c      real*8 mass_1 ! 第一段階解析用質量
c      real*8 mass_2 ! 第二段階解析用質量
c      end structure
c      record /point_s/ Point
c      ALLOCATABLE ::Point(:)
c      ALLOCATE (Point(n_point))
c
c      要素構造体
c      structure / element_s/
c      integer element_type ! 要素タイプ
c      integer n_element ! 非線形要素番号
c      real*8 E ! ヤング係数
c      real*8 G ! せん断係数
c      real*8 A ! 断面積
c      real*8 Rlx ! ねじり剛性
c      real*8 Rly ! y 軸断面二次モーメント
c      real*8 Rlz ! z 軸断面二次モーメント
c      real*8 ASy ! 各部材のY軸回りのせん断変形用等価断面積
c      real*8 ASz ! 各部材のZ軸回りのせん断変形用等価断面積
c      real*8 AM(2) ! 単位長さ当たりの質量（1:第一 2:第二段階解析用）
c      integer nm_damp ! 部材減衰の有無
c      integer nm_type ! (maxwell モデルでは、1 : x 方向 2 : y 方向 3 : z 方向)
c      integer n_section(5) ! 断面番号（修正 R0 モデルでは、データ番号）
c      integer nm_section(5) ! ファイバー数
c      real*8 ANP ! 軸方向耐力
c      real*8 AMPY ! y 軸塑性モーメント
c      real*8 AMPZ ! z 軸塑性モーメント
c      real*8 dmm(3) ! ダミー

```

```

c      real*8 i_rigid_length ! i 端剛域長さ
c      real*8 j_rigid_length ! j 端剛域長さ
c      real*8 i_shear_G ! i 端せん断剛性
c      real*8 j_shear_G ! j 端せん断剛性
c      end structure
c      record /element_s/ Element
c      ALLOCATABLE ::Element(:)
c      ALLOCATE (Element(n_element))
c
c      部材構造体
c      structure / member_s/
c      integer nm_element ! 要素番号（入力した要素番号を示す）
c      integer element_type ! 要素タイプ番号
c      integer n_model ! モデルの入れ物番号
c      integer n_model_type ! モデル別の通し番号
c      integer n_element_type ! 要素タイプ別通し番号
c      integer nm_so ! 部材の層番号
c      integer nm_dll_element ! DLL を用いた要素か（0 : システム内要素、1 : DLL 要素）
c      integer nm_point(2) ! 節点番号
c      integer irest(12) ! 部材両端の自由度番号表
c      integer ijp(2) ! 両端節点への結合状況（0:剛結合 1:ピン結合）
c      integer nm_analysis ! 部材解析種別（-1:弾性解析、その他：通常解析）
c      integer nm_group ! 部材グループ
c      integer nm_local_coord(2) ! 局所座標系の有無とその回転行列の番号
c      integer nm_damp ! 部材減衰の有無とその減衰行列の番号
c      real*8 alength ! 長さ
c      real*8 i_rigid_length ! i 端剛域長さ
c      real*8 j_rigid_length ! j 端剛域長さ
c      real*8 i_shear_G ! i 端せん断剛性
c      real*8 j_shear_G ! j 端せん断剛性
c      real*8 rot_x ! 部材主軸の回転角度（度）
c      real*8 force(12) ! 部材両端の部材端力（釣合座標系）
c      real*8 stress(18) ! 部材両端、中央の応力（部材座標系）
c      real*8 an_stress(10) ! 部材軸力（部材座標系）
c      integer d_stat(3) ! 断面の弾塑性状態（1）i 端（2）j 端（3）中央
c      end structure
c      record / member_s / Member
c      ALLOCATABLE :: Member (:)
c      ALLOCATE (Member (n_member))
c
c      ★タイトル入力
c      ierr=0
c      read(5,*,err=9911,end=9918) n
c      if(n.ne.0) then
c      write(damp_out,103)
c      103 format(//5X,'----- Title of dynamic analysis -----')
c      do j=1,n
c      read(5,'(a80)',err=9911,end=9918) title
c      write(damp_out,'(10x,a80)') title
c      end do
c      end if

```



```

c-----★制御データ入力
      read(5,*,err=9911,end=9918) node,nelem,memb,nrbound,locod,njiku
      write(damp_out,1001) '  節点数 = ',node,
*      '  要素数 = ',nelem,
*      '  部材数 = ',memb,      ! 部材数
*      '  拘束節点数= ',nrbound, ! 境界条件を与える拘束節点数
*      '  局所座標数= ',locod,   ! 局所座標を与える節点数
*      '  回転部材数= ',njiku    ! x 軸回りの回転を与える部材数
1001 format(6(/2x,a,i8))
c-----★節点データ入力
      write(damp_out,1002)
1002 format(///1h,'  節点座標'/)
      do i=1,node
      read(5,*,err=9912,end=9918) ii,x,y,z,idm
      write(damp_out,'(i4,3f12.3,i4)') ii,x,y,z,idm ! ii:節点番号 idm:ダミー (0をセット)
      Point(ii).coord(1) = x
      Point(ii).coord(2) = y
      Point(ii).coord(3) = z
      end do
c-----★節点データバッファセット(iis=node*3)
      iis=0      !実数バッファカウンタ
      jjs=0      !整数バッファカウンタ
      do i=1,node
      do j=1,3
      iis=iis+1
      buf(iis) = Point(i).coord(j)
      enddo
      enddo
c-----★節点局所座標入力
c-----★初期ゼロ設定
      do i=1,node
      Point(i).local_coord = 0
      do j=1,3
      Point(i).coord_local(j) = 0.0
      Point(i).disp_initial(j) = 0.0
      end do
      end do
c-----★節点局所座標入力
      if(locod.ne.0) then
      write(damp_out,1003)
1003 format(///1h,'  局所座標'/)
      it = 0
      do i=1,locod
      read(5,*,err=9912,end=9918) j,tlx,tly,tlz
      write(damp_out,'(i4,3f12.3)') j,tlx,tly,tlz
      it = it+1
      Point(j).local_coord = it      ! この番号は、局所座標系への変換行列の番号となる。
      Point(j).coord_local(1) = tlx
      Point(j).coord_local(2) = tly
      Point(j).coord_local(3) = tlz

```

```

      end do
      end if
c-----★局所節点データバッファセット(iis=node*3 jjs=node)
      do i=1,node
      jjs=jjs+1
      ibuf(jjs)=Point(i).local_coord
      do j=1,3
      iis=iis+1
      buf(iis) = Point(i).coord_local(j)
      enddo
      enddo
c-----★境界拘束条件入力
c-----★初期ゼロ設定
      do i=1,node
      do j=1,6
      Point(i).irest(j) = 0
      end do
      end do
c-----★境界拘束条件入力
      if(nrbound.ne.0) then
      write(damp_out,1004)
1004 format(///1h,'  境界条件'/)
      do i=1,nrbound
      read(5,*,err=9912,end=9918) j,(ire(k),k=1,6)
      write(damp_out,'(7i8)') j,(ire(k),k=1,6)
      do k=1,6
      Point(j).irest(k) = ire(k)
      end do
      end do
      end if
c-----★境界拘束条件バッファセット(jjs=6*node)
      do i=1,node
      do j=1,6
      jjs=jjs+1
      ibuf(jjs) = Point(i).irest(j)
      enddo
      enddo
c-----★線形要素モデルデータ入力
      write(damp_out,1005)
1005 format(///1h,'  要素モデルデータ'/)
      nnx=0
      do i=1,nelem
      rd1=0
      rd2=0
      sgi=0
      sg2=0
      read(5,*,err=9913,end=9918) m_type,e,g,a,rix,riy,riz,asy,asz,
*      am1,am2,anp,ampy,ampz,nm_type
      write(damp_out,'(2i4,13e12.4,i4)') i,m_type,e,g,a,rix,riy,riz,
*      asy,asz,am1,am2,anp,ampy,ampz,nm_type

```

付 分散並列型動的ソルバーの各サブルーチン

```

      Element(i).n_section(1) = 0          ! 断面におけるファイバー数をゼロセット
c-----★要素番号 13, 33 に対してデータ入力
      if(m_type .eq. 13.or.m_type .eq. 33) then
        Element(i).n_section(1) = nm_type
c      if(riy.eq.0.) riy=riz
c      if(riy.gt.riz) riy=riz          ! 弱軸の断面二次モーメントを riy にセット
      endif
c-----★要素番号 11, 15, 21 に対してデータ入力
      if(m_type .eq. 11.or.m_type .eq. 15
      * .or.m_type .eq. 21) then
        read(5,*,err=9913,end=9918) rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,2)
        write(damp_out,'(4f12.4,4i4)') rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,2)
      endif
c-----★要素番号 12, 22 に対してデータ入力
      if(m_type .eq. 12.or.m_type .eq. 22) then
        read(5,*,err=9913,end=9918) rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,3)
        write(damp_out,'(4f12.4,4i4)') rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,3)
      endif
c-----★要素番号 31 に対してデータ入力
      if(m_type .eq. 31) then
        read(5,*,err=9913,end=9918) rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,2)
        write(damp_out,'(4f12.4,4i4)') rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,2)
      endif
c-----★要素番号 32 に対してデータ入力
      if(m_type .eq. 32) then
        read(5,*,err=9913,end=9918) rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,3)
        write(damp_out,'(4f12.4,4i4)') rd1,rd2,sgi,sg2,
      * (Element(i).n_section(j),j=1,3)
      endif
c-----★新規の静的縮合モデルの第 2 レコード
      mmx=(m_type-1)/10
      if(mmx .eq. 5.or.mmx .eq. 6) then
        read(5,*,err=9913,end=9918) nxx, (n_section(j),j=1,nxx)
c-----★エレメント数チェック
c      このデータは、通常の送信データでは、データの数不定であることから送れない。
c      従って、この新規モデルを使用する場合は、E_Fiber_work(j+nnx).n_Fiber_section
c      データを用いて、別途、送信する。
c
      if(nxx.ne. S_comp_model(m_type-50).n_div_element)
      * write(76,*) ' エレメント個数エラー ', m_type, nxx
      do j=1,nxx
        E_Fiber_work(j+nnx).n_Fiber_section = n_section(j)
      enddo

```

Manual of distributed parallel processing system for dynamic analysis

```

      Element(i).n_section(1) =nnx+1      ! 新規モデルの要素内エレメントの最初の番地
      Element(i).n_section(2) =nxx        ! 新規モデルの要素内エレメントの個数,
      nnx = nnx + nxx
      endif
c-----★せん断タイプで修正 R0 モデルの数をかぞえる
      if(m_type .eq. 2) then
        if(nm_type .eq. R0_MODEL_NUMBER.or.
      * nm_type .eq. TRI_MODEL_NUMBER) then
          Model_type.n_m_ro_model= Model_type.n_m_ro_model + 1
          Element(i).n_section(1) = Model_type.n_m_ro_model
        endif
      endif
c-----★要素データを構造体にセット
      Element(i).element_type = m_type    ! 要素タイプ番号
      Element(i).E = e                   ! ヤング係数
      Element(i).G = g                   ! せん断力係数
      Element(i).A = a                   ! 断面積
      Element(i).RIx = rix                ! ねじり剛性
      Element(i).RIy = riy                ! y 軸回りの断面二次モーメント
      Element(i).RIz = riz                ! z 軸回りの断面二次モーメント
      Element(i).ASy = asy                ! y 軸に関するせん断変形に関する断面積
      Element(i).ASz = asz                ! z 軸に関するせん断変形に関する断面積
      Element(i).nm_damp = 0              ! 部材減衰有無 (システムが自動でセットする)
      Element(i).ANP = anp                ! 軸方向耐力
      Element(i).AMPY = ampy              ! y 軸塑性モーメント
      Element(i).AMPZ = ampz              ! z 軸塑性モーメント
      Element(i).nm_type = nm_type        ! 履歴タイプ番号
      Element(i).i_rigid_length = rd1     ! i 端剛域長さ
      Element(i).j_rigid_length = rd2     ! j 端剛域長さ
      Element(i).i_shear_G = sgi          ! i 端せん断剛性
      Element(i).j_shear_G = sg2          ! j 端せん断剛性
c-----★部材要素は自重を構造体にセット
      if(m_type .eq. 1.or.m_type .gt. 10) then
        Element(i).AM(1) = am1/980.      ! 要素単位立法当たり (cm*3) 質量
        Element(i).AM(2) = am2/980.      ! 要素単位立法当たり (cm*3) 質量
      else
c-----★その他はそのまの値を構造体にセット
        Element(i).AM(1) = am1
        Element(i).AM(2) = am2
      endif
      end do
c-----★要素データバッファセット (iis=nelem*17 jjs=nelem*6)
      do i=1,nelem
        buf(iis+1)=Element(i).E          ! ヤング係数
        buf(iis+2)=Element(i).G          ! せん断力係数
        buf(iis+3)=Element(i).A          ! 断面積
        buf(iis+4)=Element(i).RIx        ! ねじり剛性
        buf(iis+5)=Element(i).RIy        ! y 軸回りの断面二次モーメント
        buf(iis+6)=Element(i).RIz        ! z 軸回りの断面二次モーメント

```

付 分散並列型動的ソルバーの各サブルーチン

```

      buf(iis+7)=Element(i).ASy      ! y 軸に関するせん断変形に関する断面積
      buf(iis+8)=Element(i).ASz      ! z 軸に関するせん断変形に関する断面積

      buf(iis+9)=Element(i).ANP      ! 軸方向耐力
      buf(iis+10)=Element(i).AMPY    ! y 軸塑性モーメント
      buf(iis+11)=Element(i).AMPZ    ! z 軸塑性モーメント

      buf(iis+12)=Element(i).i_rigid_length ! i 端剛域長さ
      buf(iis+13)=Element(i).j_rigid_length ! j 端剛域長さ
      buf(iis+14)=Element(i).i_shear_G    ! i 端せん断剛性
      buf(iis+15)=Element(i).i_shear_G    ! j 端せん断剛性
c-----★部材要素は自重を構造体にセット
      buf(iis+16)=Element(i).AM(1)      ! 要素単位立法当たり (cm*3) 質量
      buf(iis+17)=Element(i).AM(2)      ! 要素単位立法当たり (cm*3) 質量
      iis=iis+17

      ibuf(jjs+1)=Element(i).element_type ! 要素タイプ番号
      ibuf(jjs+2)=Element(i).nm_damp      ! 部材減衰有無 (システムが自動でセットする)
      ibuf(jjs+3)=Element(i).nm_type      ! 履歴タイプ番号
      ibuf(jjs+4)=Element(i).n_section(1)
      ibuf(jjs+5)=Element(i).n_section(2)
      ibuf(jjs+6)=Element(i).n_section(3)
      jjs=jjs+6

      enddo

c-----★モデルタイプ別の要素数の計算 (ゼロセット)
      do i=1,Model_type.n_e_models
      Model_type.n_e_model(i)=0
      Model_type.n_m_model(i)=0
      end do
      Parameter_C.n_element_dll=0

c-----★各タイプ別の要素数を数える。
      DO i=1,nelem
      m_type = Element(i).element_type
      do j=1,Model_type.n_e_models
      if(m_type .eq. Model_type.no_e_model(j)) then
      Model_type.n_e_model(j) = Model_type.n_e_model(j)+1
      Element(i).nm_damp = Model_type.n_damp(j)
      Element(i).element_type = j      ! モデル番号からモデルの記憶領域番号に変換
      goto 19
      end if
      end do
      ierr=14
      write(damp_out,'(a,i4,a)') ' 要素:',i,
      * ' のモデルタイプが存在しない。'
      19 continue

c-----★dll を使用した要素数を数える。
      if(m_type .gt. 1000)
      * Parameter_C.n_element_dll = Parameter_C.n_element_dll+1

```

Manual of distributed parallel processing system for dynamic analysis

```

      end do
      if(ierr.ne.0) goto 9914

c-----★新規モデルの要素内エレメント数を数える
      isum=0
      do j=51,70
      if(Model_type.n_e_model(j).ne.0) then
      do k=1, Model_type.n_div_model(j)
      if(S_comp_model(j-50).nm_type_element(k).ne.1) then      ! 弾性はリエレメントを
      除く
      isum=isum+ Model_type.n_e_model(j)
      endif
      enddo
      endif
      enddo
      Model_type.n_e_New_fiber =isum      ! ②-1
      write(76,'(a,i4)') ' Model_type.n_e_New_fiber', isum

c-----★モデル別連続番号のセット (ゼロセット)
      do i=1,nelem
      Element(i).n_element=0
      enddo

c-----★
      do i=1,nelem
      iel=Element(i).element_type
      if(Element(i).n_element.eq.0) then
      ii=0
      do j=i,nelem
      if(iel.eq.Element(j).element_type) then
      ii=ii+1
      Element(j).n_element=ii
      endif
      enddo
      endif
      enddo

c-----★モデル別連続番号の出力
      write(damp_out,*) ' モデル別番号'
      do i=1,nelem
      write(damp_out,'(a,6i4)') ' element:',i,Element(i).n_element
      enddo

c-----★部材データ入力
      write(damp_out,1006)
1006 format(///1h,' 部材データ'/)
      ii1=1
      ii2=1
      ian=1
      Parameter_C.nM_New_Element = 0
      do i=1,memb
      read(5,*,err=9915,end=9918) ii,i1,i2,ie,ian,ig,iso,ii1,ii2,
      * rigid_i,rigid_j,shear_i,shear_j
      ii1=1      ! 現在 端部ピンは扱わない
      ii2=1      ! 現在 端部ピンは扱わない

```

```

Member(ii).nm_element = ie      ! 要素番号
Member(ii).nm_point(1) = i1     ! i 節点番号
Member(ii).nm_point(2) = i2     ! j 節点番号
Member(ii).nm_analysis = ian    ! 0:弾性 1:弾塑性
Member(ii).nm_group = ig       ! 部材グループ (解析に関係なし)
Member(ii).ijp(1) = ii1        ! i 節点結合状態 1:剛接合 0:ピン接合 (現在はダミー)
Member(ii).ijp(2) = ii2        ! j 節点結合状態 1:剛接合 0:ピン接合 (現在はダミー)
Member(ii).nm_dll_element = 0   ! dll 部材かどうかを示す (0:通常 1:dll 部材)
Member(ii).rot_x = 0.0         ! 主軸回転 (度) 必要ならば後からデータ入力
Member(ii).nm_so = iso         ! 部材層番号 (解析に関係なし)
Member(ii).nm_damp = 0         !
if(rigid_i.lt.0.)rigid_i=Element(ie).i_rigid_length
Member(ii).i_rigid_length=rigid_i ! i 端剛域長さ
if(rigid_j.lt.0.)rigid_j=Element(ie).j_rigid_length
Member(ii).j_rigid_length=rigid_j ! j 端剛域長さ
if(shear_i.lt.0.)shear_i=Element(ie).i_shear_G
Member(ii).i_shear_G=shear_i     ! i 端せん断剛性
if(shear_j.lt.0.)shear_j=Element(ie).j_shear_G
Member(ii).j_shear_G=shear_j     ! j 端せん断剛性
if(ie.le.nelem) then
Member(ii).element_type = Element(ie).element_type
mmx=(Member(ii).element_type-1)/10
if(mmx.eq.5.or.mmx.eq.6) then
nxx= Element(ie).n_section(2)          ! 新規モデルに含まれるエレメント数
Parameter_C.nM_New_Element = Parameter_C.nM_New_Element + nxx ! ④-1
shear_i=0
shear_j=0
elseif(Member(ii).element_type.gt.50) then
Member(ii).nm_dll_element=1
endif
write(damp_out,'(6i8,i12,2i8,4f10.2)')
*      ii,i1,i2,ie,ian,ig,iso,ii1,ii2,
*      rigid_i,rigid_j,shear_i,shear_j
else
write(damp_out,'(a,6i8)') ' data err:',ii,i1,i2,ie
endif
c-----★部材の弾塑性状態を全部材弾性にセット
do j=1,3
Member(ii).d_stat(j)=0
enddo
enddo
c-----★要素タイプ別個数チェック
do 31 i=1,nelem
j=0
do ii=1,memb
if(i.eq.Member(ii).nm_element) then
j=j+1
Member(ii).n_element_type=j
endif
enddo

```

```

31 continue
c-----★DLL 部材のチェック
Parameter_C.n_member_dll=0
DO 30 ii=1,memb
i = Member(ii).nm_element
if(i.le.nelem) then
m_type = Element(i).element_type
do j=1,Model_type.n_e_models
if(m_type.eq.j) then
Model_type.n_m_model(j) = Model_type.n_m_model(j)+1
Member(ii).n_model=j
goto 29
end if
end do
ierr=16
write(76,'(a,i4,a)') ' 部材:',i,
*      ' の要素データはモデルタイプに適合しない。'
29 continue
if(m_type.gt.1000)
*      Parameter_C.n_member_dll = Parameter_C.n_member_dll+1
else
ierr = 16
write(76,'(a,i4,a)') ' 部材:',i,
*      ' は要素データに適合しない。'
endif
30 continue
c-----★新規モデルの部材内エレメント数を数える
isum=0
do j=51,70
if(Model_type.n_e_model(j).ne.0) then
do k=1, Model_type.n_div_model(j)
if(S_comp_model(j-50).nm_type_element(k).ne.1) then ! 弾性はリエレメントを除く
isum=isum+ Model_type.n_m_model(j)
endif
enddo
endif
enddo
Model_type.n_m_New_fiber = isum ! ⑤-1
c      write(76,'(a,i4)') 'Model_type.n_m_New_fiber', isum
if(ierr.ne.0) goto 9916
c-----★モデル別通し番号計算
do 32 i=1,Model_type.n_e_models
if(Model_type.no_e_model(i).ne.0) then
j=0
do ii=1,memb
if(i.eq.Member(ii).n_model) then
j=j+1
Member(ii).n_model_type = j
endif
enddo

```

```

endif
32 continue
c-----★新規モデル 51-70 までの通し番号設定
      j=1
      do ii=1,memb
        if (Member(ii).n_model.ge.51 .and. Member(ii).n_model.le.70) then
          Member(ii).n_model_type = j
          j=j+Model_type.n_div_model(Member(ii).n_model)
        endif
      enddo

c-----★部材主軸回転入力
      if (njiku.ne.0) then
        write(damp_out,1007)
1007 format(///1h,'      部材主軸回転データ'/)
        do i=1,njiku
          read(5,*,err=9917,end=9918) ii,ax
          write(damp_out,'(i6,f12.2)') ii,ax
          Member(ii).rot_x      = ax      ! 主軸回転 (度)
        end do
      end if

c-----★部材データバッファセット (iis=memb*4 jjs=memb*14)
      do i=1,memb
        ibuf(jjs+1)=Member(i).element_type
        ibuf(jjs+2)=Member(i).nm_element      ! 要素番号
        ibuf(jjs+3)=Member(i).nm_point(1)      ! i 節点番号
        ibuf(jjs+4)=Member(i).nm_point(2)      ! j 節点番号
        ibuf(jjs+5)=Member(i).nm_analysis      ! 0:弾性 1:弾塑性
        ibuf(jjs+6)=Member(i).nm_group      ! 部材グループ (解析に関係なし)
        ibuf(jjs+7)=Member(i).ijp(1)      ! i 節点結合状態 1:剛接合 0:ピン接合 (現在は
ダミー)
        ibuf(jjs+8)=Member(i).ijp(2)      ! j 節点結合状態 1:剛接合 0:ピン接合 (現在は
ダミー)
        ibuf(jjs+9)=Member(i).nm_dll_element      ! dll 部材かどうかを示す (0:通常 1:dll 部材)
        ibuf(jjs+10)=Member(i).rot_x      ! 主軸回転 (度) 必要ならば後からデータ入力
        ibuf(jjs+11)=Member(i).nm_so      ! 部材層番号 (解析に関係なし)
        ibuf(jjs+12)=Member(i).nm_damp      !
        ibuf(jjs+13)=Member(i).nm_dll_element
        ibuf(jjs+14)=Member(i).element_type

        buf(iis+1)=Member(i).i_rigid_length      ! i 端剛域長さ
        buf(iis+2)=Member(i).j_rigid_length      ! j 端剛域長さ
        buf(iis+3)=Member(i).i_shear_G      ! i 端せん断剛性
        buf(iis+4)=Member(i).j_shear_G      ! j 端せん断剛性

        iis=iis+4
        jjs=jjs+14

      enddo
c      write(damp_out,'(a,i5,a,i5)') "iis=", iis, "jjs=", jjs

```

```

c-----★部材の両端局所座標のチェック
      do i=1,memb
        do j=1,2
          ie=Member(i).nm_point(j)
          Member(i).nm_local_coord(j) = Point(ie).local_coord
        end do
      end do

c-----★部材減衰の個数チェック
      Model_type.n_m_damp=0
      write(damp_out,'(//a)') ' 部材データ'
      write(damp_out,*) 'ii, nm_damp, nm_element, element_type',
*      'n_model, n_model_type, n_element_type'
      do ii=1,memb
        ie= Member(ii).nm_element
        if (Element(ie).nm_damp.ne.0) then
          Model_type.n_m_damp = Model_type.n_m_damp + 1
          Member(ii).nm_damp = Model_type.n_m_damp
        endif
        write(damp_out,'(i4,8i5)') ii, Member(ii).nm_damp,
*      Member(ii).nm_element, Member(ii).element_type,
*      Member(ii).n_model, Member(ii).n_model_type,
*      Member(ii).n_element_type
      end do

c-----★部材減衰の個数をセット
      Parameter_C.nc_member = Model_type.n_m_damp
      write(damp_out,'(a,i4,i4)') ' Total no. damp :',
*      Parameter_C.nc_member
      return
9911 continue
      ierr=11
      write(damp_out,'(a,a)') ' タイトル及び制御データ',
*      'にエラー、矛盾がある。'
      return
9912 continue
      ierr=12
      write(damp_out,'(a,a)') ' 節点データ、局所座標データ、',
*      '境界条件にエラーがある。'
      return
9913 continue
      ierr=13
      write(damp_out,'(a,a)') ' 要素データにエラーがある。'
      return
9914 continue
      ierr=14
      write(damp_out,'(a)') ' 要素データのモデルタイプにエラーがある。'
      return
9915 continue
      ierr=15
      write(damp_out,'(a,a)') ' 部材データにエラーがある。'
      return

```

```

9916 continue
      ierr=16
      write(damp_out,'(a,a)') ' 部材データに矛盾がある。'
      return
9917 continue
      ierr=17
      write(damp_out,'(a,a)') ' 主軸回転データにエラーがある。'
      return
9918 continue
      ierr=18
      write(damp_out,'(a,a)') ' 構造データファイルが不足している。'
      return
      end

```

4. set_structure

```

C
C  ● SUBROUTINE /set_structure
C
C  ● 構造データをマスターより受信し、セットする
C
      subroutine set_structure(Point, Member, Element, Parameter_C,
*          Model_type, ierr, buf, ibuf, n_member1, n_member2,
*          S_comp_model, E_Fiber_work)
C-----★計算結果のダンプ出力ファイル番号
      parameter(damp_out = 76)
      implicit real*8(A-H, O-Z)
      include "..¥..¥sf3st¥submain.h"
      include "..¥..¥sf3st¥New_submain.h"
      record / E_Fiber_work_s / E_Fiber_work
      record / S_comp_model_s / S_comp_model
      record / parameter_s    / Parameter_C
      record / member_s       / Member
      record / point_s        / Point
      record / element_s      / Element
      record / n_model_s       / Model_type
      dimension Member(*), Point(*), Element(*)
      dimension n_section(100), E_Fiber_work(*), S_comp_model(*)
      character title*80
      dimension ire(6)          ! ワークエリアとして使用
      integer RO_MODEL_NUMBER, TRI_MODEL_NUMBER

C
      integer ibuf(*)
      dimension buf(*)

C
      data          RO_MODEL_NUMBER/12/
      data          TRI_MODEL_NUMBER/11/

C
      バッファデータ仕様
      buf()      1 : 座標 3
      buf()      2 : 局所座標 3

```

```

C          3 : 要素 17
C          4 : 部材 4
C      ibuf()  1 : 境界拘束条件 6
C          2 : 要素 6
C          3 : 部材 14
C
C-----
C
C      Parameter_C      : 構造体
C      Member           : 構造体
C      Element          : 構造体
C      Point            : 構造体
C      Mode_type        : 構造体
C      構造体の定義は、インクルードファイル「submain.h」で行われている。
C
C
C      節点構造体
C      structure / point_s/
C      real*8      coord(3)          ! 3次元の節点座標
C      integer      irest(6)         ! 節点自由度の拘束表
C      integer      local_coord      ! 局所座標系の有無（ある場合は、回転行列の通し番号）
C      real*8      coord_local(3)    ! 局所座標系（全体座標系に対する角度）
C      real*8      disp_initial(3)   ! 初期変位
C      real*8      mass_1            ! 第一ステップ質量
C      real*8      mass_2            ! 第二ステップ質量
C      end structure
C      record /point_s/ Point
C      ALLOCATABLE ::Point(:)
C      ALLOCATE (Point(n_point))
C
C      要素構造体
C      structure / element_s/
C      integer      element_type     ! 要素タイプ
C      integer      n_element        ! 非線形要素番号
C      real*8      E                 ! ヤング係数
C      real*8      G                 ! せん断係数
C      real*8      A                 ! 断面積
C      real*8      RIx               ! ねじり剛性
C      real*8      RIy               ! y 軸断面二次モーメント
C      real*8      RIz               ! z 軸断面二次モーメント
C      real*8      ASy               ! 各部材のY 軸回りのせん断変形用等価断面積
C      real*8      ASz               ! 各部材のZ 軸回りのせん断変形用等価断面積
C      real*8      AM(2)             ! 単位長さ当たりの質量（1:第一 2:第二ステップ用）
C      integer      nm_damp           ! 部材減衰の有無
C      integer      nm_type           ! (maxwell モデルでは、1 : x 方向 2 : y 方向 3 : z 方向)
C      integer      n_section(5)     ! 断面番号（修正 R0 モデルでは、データ番号）
C      integer      nm_section(5)    ! ファイバー数
C      real*8      ANP               ! 軸方向耐力
C      real*8      AMPY              ! y 軸塑性モーメント
C      real*8      AMPZ              ! z 軸塑性モーメント
C      real*8      dmm(3)            ! ダミー

```

付 分散並列型動的ソルバーの各サブルーチン

```

c      real*8   i_rigid_length ! i 端剛域長さ
c      real*8   j_rigid_length ! j 端剛域長さ
c      real*8   i_shear_G      ! i 端せん断剛性
c      real*8   j_shear_G      ! j 端せん断剛性
c      end structure
c      record /element_s/ Element
c      ALLOCATABLE ::Element(:)
c      ALLOCATE (Element(n_element))
c
c      部材構造体
c      structure / member_s/
c      integer nm_element      ! 要素番号 (入力した要素番号を示す)
c      integer element_type    ! 要素タイプ番号
c      integer n_model         ! モデルの入れ物番号
c      integer n_model_type    ! モデル別の通し番号
c      integer n_element_type  ! 要素タイプ別通し番号
c      integer nm_so           ! 部材の層番号
c      integer nm_dll_element  ! DLL を用いた要素か (0 : システム内要素、1 : DLL 要素)
c      integer nm_point(2)     ! 節点番号
c      integer irest(12)       ! 部材両端の自由度番号表
c      integer ijp(2)          ! 両端節点への結合状況 (0:剛結合 1:ピン結合)
c      integer nm_analysis     ! 部材解析種別 (-1:弾性解析、その他:通常解析)
c      integer nm_group        ! 部材グループ
c      integer nm_local_coord(2) ! 局所座標系の有無とその回転行列の番号
c      integer nm_damp         ! 部材減衰の有無とその減衰行列の番号
c      real*8   alength        ! 長さ
c      real*8   i_rigid_length ! i 端剛域長さ
c      real*8   j_rigid_length ! j 端剛域長さ
c      real*8   i_shear_G      ! i 端せん断剛性
c      real*8   j_shear_G      ! j 端せん断剛性
c      real*8   rot_x           ! 部材主軸の回転角度 (度)
c      real*8   force(12)      ! 部材両端の部材端力 (釣合座標系)
c      real*8   stress(18)     ! 部材両端、中央の応力 (部材座標系)
c      real*8   an_stress(10)  ! 部材軸力 (部材座標系)
c      integer d_stat(3)       ! 断面の弾塑性状態 (1) i 端 (2) j 端 (3) 中央
c      end structure
c      record / member_s / Member
c      ALLOCATABLE :: Member(:)
c      ALLOCATE (Member (n_member))
c
c      node=Parameter_C.n_point
c      nelem=Parameter_C.n_element
c      memb=Parameter_C.n_member
c      nrbound=Parameter_C.n_boundary_p
c      locod=Parameter_C.n_local_coord
c      njiku=Parameter_C.n_rot_axis
1001 format(6//2x,a,i8))
      write(damp_out,1001) '  節点数 = ',node,
*                          '  要素数 = ',nelem,
*                          '  部材数 = ',memb,          ! 部材数

```

Manual of distributed parallel processing system for dynamic analysis

```

*      ' 拘束節点数= ',nrbound,      ! 境界条件を与える拘束節点数
*      ' 局所座標系= ',locod,        ! 局所座標を与える節点数
*      ' 回転部材数= ',njiku
c      ★節点データセット
c
c      iis=0
c      jjs=0
c      do i=1,node
c      do j=1,3
c      iis=iis+1
c      Point(i).coord(j)=buf(iis)
c      enddo
c      enddo
c
c      ★局所節点データバッファセット
c
c      do i=1,node
c      jjs=jjs+1
c      Point(i).local_coord=ibuf(jjs)
c      do j=1,3
c      iis=iis+1
c      Point(i).coord_local(j)=buf(iis)
c      enddo
c      enddo
c
c      ★境界拘束条件バッファセット
c
c      write(76,'(/a,i4)') '  節点情報 ',node
c      do i=1,node
c      do j=1,6
c      jjs=jjs+1
c      Point(i).irest(j)=ibuf(jjs)
c      enddo
c      enddo
c      do i=1,node
c      write(damp_out,'(i4,6f12.3/6i4)')
*      i, (Point(i).coord(j), j=1,3),
*      (Point(i).coord_local(j), j=1,3),
*      (Point(i).irest(j), j=1,6)
c      enddo
c
c      ★要素データバッファセット
c
c      write(76,'(/a,i4)') '  要素情報 ',nelem
c      nnx=0
c      do i=1,nelem
c      Element(i).E=buf(iis+1)          ! ヤング係数
c      Element(i).G=buf(iis+2)          ! せん断力係数
c      Element(i).A=buf(iis+3)          ! 断面積
c      Element(i).Rlx=buf(iis+4)        ! ねじり剛性
c      Element(i).Rly=buf(iis+5)        ! y 軸回りの断面二次モーメント
c      Element(i).Rlz=buf(iis+6)        ! z 軸回りの断面二次モーメント
c      Element(i).ASy=buf(iis+7)        ! y 軸に関するせん断変形に關係する断面積
c      Element(i).ASz=buf(iis+8)        ! z 軸に関するせん断変形に關係する断面積
c
c      Element(i).ANP=buf(iis+9)        ! 軸方向耐力

```

```

Element(i).AMPY=buf(iis+10)      ! y 軸塑性モーメント
Element(i).AMPZ=buf(iis+11)      ! z 軸塑性モーメント

Element(i).i_rigid_length=buf(iis+12) ! i 端剛域長さ
Element(i).j_rigid_length=buf(iis+13) ! j 端剛域長さ
Element(i).i_shear_G=buf(iis+14)    ! i 端せん断剛性
Element(i).j_shear_G=buf(iis+15)    ! j 端せん断剛性
c-----★部材要素は自重を構造体にセット
Element(i).AM(1)=buf(iis+16)      ! 要素単位立法当たり (cm*3) 質量
Element(i).AM(2)=buf(iis+17)      ! 要素単位立法当たり (cm*3) 質量
iis=iis+17

Element(i).element_type=ibuf(jjs+1) ! 要素タイプ番号
Element(i).nm_damp=ibuf(jjs+2)      ! 部材減衰有無 (システムが自動でセットする)
Element(i).nm_type=ibuf(jjs+3)      ! 履歴タイプ番号
Element(i).n_section(1)=ibuf(jjs+4)
Element(i).n_section(2)=ibuf(jjs+5)
Element(i).n_section(3)=ibuf(jjs+6)

write(76,'(2i4,3f10.2)') i,Element(i).element_type,
* Element(i).E,Element(i).G,Element(i).A

jjs=jjs+6

m_type=Element(i).element_type
nm_type=Element(i).nm_type
c write(76,'(2i4)') m_type,nm_type
c-----★新規の静的縮合モデルの第2レコード
mmx=(m_type-1)/10
if(mmx.eq.5.or.mmx.eq.6) then
c-----★エレメント数チェック
nxx=S_comp_model(m_type-50).n_div_element
Element(i).n_section(1)=nnx+1      ! 新規モデルの要素内エレメントの最初の番地
Element(i).n_section(2)=nnx        ! 新規モデルの要素内エレメントの個数,
nnx=nnx+nnx
c write(76,'(a,10i4)') ' m_type ',nnx,nxx,m_type,mmx
endif
c-----★せん断タイプで修正 R0 モデルの数をかぞえる
if(m_type.eq.2) then
if(nm_type.eq.R0_MODEL_NUMBER.or.
* nm_type.eq.TRI_MODEL_NUMBER) then
Model_type.n_m_ro_model=Model_type.n_m_ro_model+1
Element(i).n_section(1)=Model_type.n_m_ro_model
endif
endif

end do

c-----★モデルタイプ別の要素数の計算 (ゼロセット)
do i=1,Model_type.n_e_models

```

```

Model_type.n_e_model(i)=0
Model_type.n_m_model(i)=0
end do
Parameter_C.n_element_dll=0
c-----★各タイプ別の要素数を数える。
DO i=1,nelem
m_type = Element(i).element_type
do j=1,Model_type.n_e_models
if(m_type.eq.Model_type.no_e_model(j)) then
Model_type.n_e_model(j)=Model_type.n_e_model(j)+1
Element(i).nm_damp=Model_type.n_damp(j)
Element(i).element_type=j      ! モデル番号からモデルの記憶領域番号に変換
c write(76,'(a,3i4)') ' model check ',m_type,j,
c * Model_type.no_e_model(j)
goto 19
end if
end do
ierr=14
c write(damp_out,'(a,i4,a)') ' 要素:',i,
c * ' のモデルタイプが存在しない。'
19 continue
c-----★dllを使用した要素数を数える。
if(m_type.gt.1000)
* Parameter_C.n_element_dll=Parameter_C.n_element_dll+1
end do
c if(ierr.ne.0) goto 9914
c-----★新規モデルの要素内エレメント数を数える
isum=0
do j=51,70
if(Model_type.n_e_model(j).ne.0) then
do k=1,Model_type.n_div_model(j)
if(S_comp_model(j-50).nm_type_element(k).ne.1) then      ! 弾性はりエレメントを
除く
isum=isum+Model_type.n_e_model(j)
endif
enddo
endif
enddo
Model_type.n_e_New_fiber=isum
c-----★モデル別連続番号のセット (ゼロセット)
do i=1,nelem
Element(i).n_element=0
enddo
c-----★
do i=1,nelem
iel=Element(i).element_type
if(Element(i).n_element.eq.0) then
ii=0
do j=i,nelem
if(iel.eq.Element(j).element_type) then

```



```

      ii=ii+1
      Element(j).n_element=ii
    endif
  enddo
endif
enddo

c-----★モデル別連続番号の出力
      write(damp_out,' (/a)') ' モデル別番号'
      do i=1,nelem
        write(damp_out,' (a,6i4)') ' element:',i,Element(i).element_type,
* Element(i).n_element
      enddo

c-----★部材データバッファセット
      i=0
      Parameter_C.nM_New_Element = 0
c      write(76,' (a,6i4)') ' membs ',memb,n_member1,n_member2
      do js=1,memb

c-----★担当部材以外を削除
      if(js.lt.n_member1.or.js.gt.n_member2) then
        iis=iis+4
        jjs=jjs+14
      else
        i=i+1
        Member(i).element_type=ibuf(jjs+1)
        Member(i).nm_element=ibuf(jjs+2)      ! 要素番号
        Member(i).nm_point(1)=ibuf(jjs+3)     ! i 節点番号
        Member(i).nm_point(2)=ibuf(jjs+4)     ! j 節点番号
        Member(i).nm_analysis=ibuf(jjs+5)     ! 0:弾性 1:弾塑性
        Member(i).nm_group=ibuf(jjs+6)        ! 部材グループ (解析に 関係なし)
        Member(i).ijp(1)=ibuf(jjs+7)          ! i 節点結合状態 1:剛接合 0:ピン 接合 (現在
はダミー)
        Member(i).ijp(2)=ibuf(jjs+8)          ! i 節点結合状態 1:剛接合 0:ピン 接合 (現在
はダミー)
        Member(i).nm_dll_element=ibuf(jjs+9)  ! dll 部材かどうかを示す (0:通常 1:dll 部材)
        Member(i).rot_x=ibuf(jjs+10)          ! 主軸回転 (度) 必要ならば後からデータ入力
        Member(i).nm_so=ibuf(jjs+11)          ! 部材層番号 (解析に 関係なし)
        Member(i).nm_damp=ibuf(jjs+12)        !
        Member(i).nm_dll_element=ibuf(jjs+13)
        Member(i).element_type=ibuf(jjs+14)   ! ダミー

        Member(i).i_rigid_length=buf(iis+1)   ! i 端剛域長さ
        Member(i).j_rigid_length=buf(iis+2)   ! j 端剛域長さ
        Member(i).i_shear_G=buf(iis+3)        ! i 端せん断剛性
        Member(i).j_shear_G=buf(iis+4)        ! j 端せん断剛性
        ii=i
        mmx=(Member(i).element_type-1)/10
        if(mmx.eq.5.or.mmx.eq.6) then
          ie=Member(i).nm_element
          nxx= Element(ie).n_section(2)        ! 新規モデルに含まれるエレメント数
          Parameter_C.nM_New_Element = Parameter_C.nM_New_Element + nxx      ! ④-1

```

```

c      write(76,' (a,10i4)') ' nxx :',mmx,nxx,ie
      elseif(Member(i).element_type.gt.50) then
        Member(i).nm_dll_element=1
      endif

      write(damp_out,' (2i4,8i5)') js,ii,Member(i).element_type,
* Member(ii).nm_element,Member(ii).nm_point(1),
* Member(ii).nm_point(2),Member(ii).nm_analysis,
* Element(1).element_type

      iis=iis+4
      jjs=jjs+14

c-----★部材の弾塑性状態を全部材弾性にセット
      do j=1,3
        Member(i).d_stat(j)=0
      enddo
    endif

    end do
    memb=i      !担当部材数をセット (以後のチェックは担当部材で行う)
    Parameter_C.n_member=memb

c-----★要素タイプ別個数チェック
      do 31 i=1,nelem
        j=0
        do ii=1,memb
          if(i.eq.Member(ii).nm_element) then
            j=j+1
            Member(ii).n_element_type=j
          c      write(76,' (a,6i5)') ' element ',i,ii,j
          endif
        enddo
        31 continue

c-----★DLL 部材のチェック
      write(76,' (/a,i4)') ' 部材情報',memb
      Parameter_C.n_member_dll=0
      DO 30 ii=1,memb
        i = Member(ii).nm_element
        if(i.le.nelem) then
          m_type = Element(i).element_type
          do j=1,Model_type.n_e_models
            if(m_type.eq.j) then
              Model_type.n_m_model(j) = Model_type.n_m_model(j)+1
              Member(ii).n_model= j
            goto 29
          end if
        end do
        29 continue
        if(m_type.gt.1000)
          * Parameter_C.n_member_dll = Parameter_C.n_member_dll+1

```

```

      else
      ierr = 16
      endif
30  continue
c    if(ierr.ne.0) goto 9916
c-----★新規モデルの部材内エレメント数を数える
      isum=0
      do j=51,70
      if(Model_type.n_e_model(j).ne.0) then
      do k=1, Model_type.n_div_model(j)
      if(S_comp_model(j-50).nm_type_element(k).ne.1) then      ! 弾性はリエレメントを除く
      isum=isum+ Model_type.n_m_model(j)
      endif
      enddo
      endif
      enddo
      Model_type.n_m_New_fiber =isum      ! ⑤-1
c    write(76,'(a,i4)') ' Model_type.n_m_New_fiber', isum
c    if(ierr.ne.0) goto 9916
c-----★モデル別通し番号計算
      do 32 i=1, Model_type.n_e_models
      if(Model_type.no_e_model(i).ne.0) then
      j=0
      do ii=1,memb
      if(i.eq.Member(ii).n_model) then
      j=j+1
      Member(ii).n_model_type = j
      endif
      enddo
      endif
32  continue
c-----★新規モデル 51-70 までの通し番号設定
      j=1
      do ii=1,memb
      if(Member(ii).n_model.ge.51 .and. Member(ii).n_model.le.70) then
      Member(ii).n_model_type = j
      j=j+Model_type.n_div_model(Member(ii).n_model)
      endif
      enddo
c-----★部材の両端局所座標のチェック
      do i=1,memb
      do j=1,2
      ie=Member(i).nm_point(j)
      Member(i).nm_local_coord(j) = Point(ie).local_coord
      end do
      end do
c-----★部材減衰の個数チェック
      Model_type.n_m_damp=0
      do ii=1,memb

```

```

      ie= Member(ii).nm_element
      if(Element(ie).nm_damp.ne.0) then
      Model_type.n_m_damp = Model_type.n_m_damp + 1
      Member(ii).nm_damp = Model_type.n_m_damp
      endif
      write(damp_out,'(i4,9i5)') ii, Member(ii).nm_damp,
      *      Member(ii).nm_element, Member(ii).element_type,
      *      Member(ii).n_model, Member(ii).n_model_type,
      *      Member(ii).n_element_type
      end do
c-----★部材減衰の個数をセット
      Parameter_C.nc_member = Model_type.n_m_damp
      write(damp_out,'(a,i4,i4)') ' Total no. damp :',
      *      Parameter_C.nc_member

      return
      end

```

5. Get_pointforce_ld_pa

```

C
C
C ● SUBROUTINE /Get_pointforce_ld_pa
C
C ● 部材節点力のセット(ok)
C
subroutine Get_pointforce_ld_pa(n_member1,n_member2,
*      ld_point, Member)
implicit real*8(A-H,0-Z)
include "..¥..¥sf3st¥submain.h"
record / member_s / Member
dimension Member(*)
real*8 ld_point(*)
C
C      ld_point      :real*8 右辺項
C      Member        :structure
C      n_member       :integer 部材数
C
do i=n_member1,n_member2
do j=1,12
C
C      Maxwell Model の応力は、他で考慮するのでここでは、無視する。
if(Member(i).element_type.ne.6) then
i1 = Member(i).irest(j)
if(i1.gt.0) ld_point(i1)=ld_point(i1) - Member(i).force(j)
endif
end do
c write(76,'(i3,12e10.3)') i, (Member(i).force(j), j=1,12)
end do
return
end

```

付 2.2 部材関連サブルーチン

ここで掲載するサブルーチンは、分担部材に関する変更部分を含むものである。変更部分は比較的少ないのが特徴であり、これらサブルーチンの内容については、「動的解析編」を参照されたい。

1. Add_damp2_ld_pa

```

C
C  ● SUBROUTINE /Add_damp2_ld_pa (チェック OK)
C
C  ● 減衰・整合質量による減衰項のセット(ok)
C
subroutine Add_damp2_ld_pa(n_member1,n_member2,nx,ld_point,Member,
*   past_disp_point,past_vel_point,past_acc_point,
*   am_member,Newmark_P,Element,load_mass)
implicit real*8(A-H,O-Z)
include "..¥..¥sf3st¥submain.h"
record / newmark_s    /Newmark_P
record / member_s     /Member
record / element_s    /Element
dimension Member(*),Element(*)
real*8 ld_point(*),am_member(12,12,*)
dimension u(12)
dimension past_disp_point(*),past_vel_point(*),past_acc_point(*)

C
C  nx                :integer   第一ステップか第二ステップか
C  ld_point(*)       :real*8    右辺荷重項
C  Point             :structure
C  n_point           :integer   節点数
C  past_disp_point   :real*8    変位
C  past_vel_point    :real*8    速度
C  past_acc_point    :real*8    加速度
C  am_point(2,n_point) :real*8  部材整合質量 (釣合系)
C  Newmark_P         :structure
C  Parameter_C       :structure
C  Element           :structure
C  load_mass         :integer   整合質量を用いるかどうか

if(load_mass.eq.0) return

if(nx.eq.1) then
a= Newmark_P.alf1_1
b= Newmark_P.alf1_1*Newmark_P.ddt_1
ik=1
else
a= Newmark_P.alf2_1
b= Newmark_P.alf2_1*Newmark_P.ddt_1
ik=2

```

endif

```

do i=n_member1,n_member2
ie = Member(i).nm_element
if(Element(ie).am(ik).ne.0.0) then
do j=1,12
irest = Member(i).irest(j)
if(irest.gt.0) then
u(j)=a*past_vel_point(irest) +
*   b*past_acc_point(irest)
else
u(j)=0.0
endif
endif
end do

```

```

do j=1,12
i1=Member(i).irest(j)
if(i1.gt.0) then
sum=0.
do k=1,12
sum=sum+am_member(j,k,i)*u(k)
enddo
ld_point(i1)=ld_point(i1) - sum
endif
endif
end do

```

```

endif
end do

```

```

return
end

```

2. Add_damp3_ld_pa

```

C
C  ● SUBROUTINE /Add_damp3_ld_pa (チェック OK)
C
C  ● 部材減衰による減衰項のセット(ok)
C
subroutine Add_damp3_ld_pa(n_member1,n_member2,nx,ld_point,Member,
*   past_disp_point,past_vel_point,past_acc_point,
*   ac_member,Newmark_P,n_damp)

implicit real*8(A-H,O-Z)
include "..¥..¥sf3st¥submain.h"
record / newmark_s    /Newmark_P
record / member_s     /Member
dimension Member(*)
real*8 ld_point(*),ac_member(12,12,*)
dimension u(12)
dimension past_disp_point(*),past_vel_point(*),past_acc_point(*)

```

```

c
c      nx                      :integer  第一ステップか第二ステップか
c      ld_point(*)            :real*8    右辺荷重項
c      Point                   :structure
c      n_point                 :integer  節点数
c      past_disp_point         :real*8    変位
c      past_vel_point          :real*8    速度
c      past_acc_point          :real*8    加速度
c      ac_point(2,n_point)     :real*8    部材減衰行列 (釣合系)
c      Newmark_P               :structure
c      Parameter_C             :structure
c      n_damp                   :integer  部材減衰の部材がありか
c
c      if(n_damp.eq.0) return

      b=Newmark_P.ddt_1
      do i=n_member1,n_member2
        ij = Member(i).nm_damp
        if(ij.ne.0) then
          do j=1,12
            irest = Member(i).irest(j)
            if(irest.gt.0) then
              u(j)= past_vel_point(irest) +
*              b*past_acc_point(irest)
            else
              u(j)=0.0
            endif
          end do
          do j=1,12
            irest=Member(i).irest(j)
            if(irest.gt.0) then
              sum=0.
              do k=1,12
                sum=sum+ac_member(j,k,ij)*u(k)
              enddo
              ld_point(irest)=ld_point(irest) - sum
            endif
          end do
        endif
      end do

      return
      end

```

3. Add_earth2_ld_pa

```

c
c      ● SUBROUTINE /Add_earth2_ld_pa
c
c      ● 整合質量質量行列に関する地震荷重(ok)
c

```

```

      subroutine Add_earth2_ld_pa(n_member1,n_member2,acc1,acc2,acc3,
*      ld_point,Member,am_member,rot_local,Parameter_C,Dynamic_load)
      implicit real*8(A-H,O-Z)
      include "..¥..¥sf3st¥submain.h"
      dimension am_member(12,12,*)
      dimension rot_local(3,3,*)
      record / member_s      / Member
      record / parameter_s   / Parameter_C
      record / dynamic_load_s/ Dynamic_load
      dimension Member(*)
      dimension amk(12),amkk(6)
      real*8 ld_point(*)
c
c      n_istep                  :integer  第一ステップか第二ステップか
c      acc1,acc2,acc3          :real*8    x、y、z方向の地震荷重
c      ld_point(*)             :real*8    右辺荷重項
c      Point                   :structure
c      n_point                 :integer  節点数
c      am_point(2,n_point)     :real*8    節点集中質量
c      rot_local(3,3,*)        :real*8    全体座標系から局所座標への回転行列
c      Parameter_C             :structure
c
c      if(Dynamic_load.load_mass.eq.0) return

      do i=1,12
        amk(i)=0.0
      enddo
      if(Parameter_C.n_local_coord.ne.0) then
        do i=n_member1,n_member2
          do j=1,12
            amk(j)=acc1*am_member(j,1,i)+acc1*am_member(j,7,i)
*            +acc2*am_member(j,2,i)+acc2*am_member(j,8,i)
*            +acc3*am_member(j,3,i)+acc3*am_member(j,9,i)
          enddo
          do k=1,2
            ij=Member(i).nm_local_coord(k)
            if(ij.eq.0) then
              do j=1,6
                j1=j+(k-1)*6
                i1=Member(i).irest(j1)
                if(i1.gt.0) ld_point(i1)=ld_point(i1) - amk(j1)
              end do
            else
              call RotateTLss_v(2,amk(1+(k-1)*6),rot_local(1,1,ij),amkk)
              do j=1,6
                j1=j+(k-1)*6
                i1=Member(i).irest(j1)
                if(i1.gt.0) ld_point(i1)=ld_point(i1) - amkk(j)
              end do
            end if
          end do
        end do
      end if

```

```

endif
end do
end do

else
do i=n_member1,n_member2
do j=1,12
amk(j)=acc1*am_member(j,1,i)+acc1*am_member(j,7,i)
*      +acc2*am_member(j,2,i)+acc2*am_member(j,8,i)
*      +acc3*am_member(j,3,i)+acc3*am_member(j,9,i)
enddo
do j=1,12
il=Member(i).irest(j)
if(il.gt.0) ld_point(il)=ld_point(il) - amk(j)
end do
end do

end if
return
end

```

4. Add_fdd_ld_pa

C

C ● SUBROUTINE /Add_fdd_ld_pa Parallel version (チェック OK)

C

C ● Maxwell 減衰項に関するベクトルを加える。(ok)

C

```

subroutine Add_fdd_ld_pa(n_member1,n_member2,ld_point_repeat,
*      E_model6_real,Element,
*      Member,est_vel_point,rot_memb)
implicit real*8(A-H,O-Z)
include "..\%.%sf3st%submain.h"
record / member_s / Member
record / e_model6_real_s / E_model6_real
record / element_s / Element
dimension Member(*),E_model6_real(*),Element(*)
real*8 ld_point_repeat(*),est_vel_point(*)
dimension rot_memb(3,3,2,*)
dimension av(12),ud(12),vpp(12)

```

C

C ld_point_repeat(*) : real*8 線形右辺項ベクトル

C Member : structure

C n_member : integer 部材数

C est_vel_point : real*8 予測節点速度

C

```

do i=n_member1,n_member2
mem = i
iet = Member(i).element_type
iet=(iet-1)/10
ie = Member(i).nm_element

```

C ★部材減衰を持っているか

```

if( Element(ie).nm_damp.ne. 0) then
ien= Member(i).n_model_type
write(76,'(a,3i4)') ' mem:',i,Member(i).nm_point(1),
*      Member(i).nm_point(2)
do j=1,12
irest = Member(i).irest(j)
if(irest.ne.0) then
ud(j)=est_vel_point(irest)
else
ud(j)=0.
endif
enddo
c write(76,'(6e12.4,3x,6e12.4)') ( ud(j),j=1,12)
call RotateL_v(1,ud,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
c write(76,'(6e12.4,3x,6e12.4)') ( vpp(j),j=1,12)
if(Member(i).nm_dll_element.ne. 0) goto 9999 ! DLL 要素
if(iett.eq.0)then
goto(11,12,13,14,15,16,17,18,19,20), iet
11 continue

```

C ★Model_No.1 通常の有限要素弾塑性モデル

goto 100

12 continue

C ★Model_No.2 3次元せん断弾塑性モデル

goto 100

13 continue

C ★Model_No.3 3次元軸力弾塑性モデル

goto 100

14 continue

C ★Model_No.4 3次元ケーブル弾塑性モデル

goto 100

15 continue

C ★Model_No.5 3次元免振モデル

goto 100

16 continue

C ★Model_No.6 3次元制震 Maxwell モデル

```

ii=Element(ie).nm_type
call Cal_force_maxwelldamp(vpp,E_model6_real(ien),av,ii,i)
goto 100
17 continue

```

C ★Model_No.7 3次元プレテンション動作モデル

```

      goto 100
      18 continue
c-----★Model_No. 8

      goto 100
      19 continue
c-----★Model_No. 9

      goto 100
      20 continue
c-----★Model_No. 10

      goto 100

      elseif(iett.eq.1)then
      goto(111,112,113,114,115,116,117,118,119,120), iet-10
      111 continue
c-----★Model_No. 11 両端ファイバーモデル

      goto 100
      112 continue
c-----★Model_No. 12 両端、中央ファイバーモデル

      goto 100
      113 continue
c-----★Model_No. 13 両端 MS モデル

      goto 100
      114 continue
c-----★Model_No. 14 両端、中央 MS モデル

      goto 100
      115 continue
c-----★Model_No. 15 幾何学非線形+弾塑性型有
要素モデル

      goto 100
      116 continue
c-----★Model_No. 16 3次元プレテンション動作
モデル

      goto 100
      117 continue
c-----★Model_No. 17

      goto 100
      118 continue
c-----★Model_No. 18

```

```

      goto 100
      119 continue
c-----★Model_No. 19

      goto 100
      120 continue
c-----★Model_No. 20

      goto 100
      endif

9999 continue

c-----★Model_No. DLL
c      call Cal_lin_damp_dll(mem,
c      * work1_element,work2_element,work1_member,work2_member)

      100 continue
c-----★右変更への追加

      call RotateL_v(2,av,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vpp)
      do j=1,12
      irest = Member(i).irest(j)
      if(irest.ne.0) then
      ld_point_repeat(irest)=ld_point_repeat(irest) - vpp(j)
      end if
      end do
      endif
      end do

      return
      end

```

5. Add_stiff1_ld_pa

```

c
c      ● SUBROUTINE /Add_stiff1_ld_pa (チェック OK)
c
c      ● 線形剛性項に関するベクトルを加える。その1 (ok)
c
      subroutine Add_stiff1_ld_pa(n_member1,n_member2,n_istep,ld_point,
      * Member,past_disp_point,past_vel_point,past_acc_point,
      * ak_linear,Newmark_P)
      implicit real*8(A-H,O-Z)
      include "..¥..¥sf3st¥submain.h"
      record / member_s / Member
      record / newmark_s / Newmark_P
      dimension Member(*)
      real*8 ld_point(*),past_disp_point(*)
      dimension past_vel_point(*),past_acc_point(*)
      dimension u(12),ak_linear(12,12,*)
c

```

付 分散並列型動的ソルバーの各サブルーチン

```

c      n_istep          : integer   第一ステップか第二ステップか
c      ld_point(*)      : real*8    線形右辺項ベクトル
c      Member           : structure
c      n_member         : integer   部材数
c      past_disp_point  : real*8    Δ t 秒前の節点変位
c      past_vel_point   : real*8    Δ t 秒前の節点速度
c      past_acc_point   : real*8    Δ t 秒前の節点加速度
c      ak_linear(*)     : real*8    線形剛性行列
c      Newmark_P        : structure

```

```

c      if(n_istep.eq.1) then
c          剛性          減衰
a=Newmark_P.dt      + Newmark_P.alf1_2
b=Newmark_P.bdt_5 + Newmark_P.alf1_2*Newmark_P.ddt_1
else
a=Newmark_P.dt      + Newmark_P.alf2_2
b=Newmark_P.bdt_5 + Newmark_P.alf2_2*Newmark_P.ddt_1
endif

do i=n_member1,n_member2
do j=1,12
irest = Member(i).irest(j)
if(irest.ne.0) then
u(j)=past_disp_point(irest)+a*past_vel_point(irest) +
*      b*past_acc_point(irest)
else
u(j)=0.0
endif
end do
do j=1,12
irest = Member(i).irest(j)
if(irest.ne.0) then
sum=0.0
do k=1,12
sum=sum+ak_linear(j,k,i)*u(k)
end do
ld_point(irest)=ld_point(irest) - sum
end if
end do

end do
return
end

```

6. Add_stiff2_ld_pa

```

c
c      ● SUBROUTINE /Add_stiff2_ld_pa Parallel version (チェック OK)
c
c      ● 線形剛性項に関するベクトルを加える。その2 (ok)
c

```

Manual of distributed parallel processing system for dynamic analysis

```

subroutine Add_stiff2_ld_pa(n_member1,n_member2,ld_point_repeat,
*      Member,n_member,est_disp_point,ak_linear)
implicit real*8(A-H,O-Z)
include "..¥..¥sf3st¥submain.h"
record / member_s / Member
dimension Member(n_member)
real*8 ld_point_repeat(*),est_disp_point(*)
dimension u(12),ak_linear(12,12,*)

```

```

c
c      ld_point_repeat(*) : real*8    線形右辺項ベクトル
c      Member             : structure
c      n_member           : integer   部材数
c      est_disp_point     : real*8    予測節点変位
c      ak_linear(*)       : real*8    線形剛性行列
c

```

```

do i=n_member1,n_member2
do j=1,12
irest = Member(i).irest(j)

if(irest.gt.0) then
u(j)=est_disp_point(irest)
else
u(j)=0.0
endif
end do

do j=1,12
irest = Member(i).irest(j)
if(irest.gt.0) then
sum=0.0
do k=1,12
sum=sum+ak_linear(j,k,i)*u(k)
end do
ld_point_repeat(irest)=ld_point_repeat(irest)+sum
end if
end do
end do
return
end

```

7. Add_tan_stiff_ld_pa

```

c
c      ● SUBROUTINE /Add_tan_stiff_ld (チェック OK)
c
c      ● 接線剛性項に関するベクトルを加える。その2 (ok)
c

```

```

subroutine Add_tan_stiff_ld_pa(n_member1,n_member2,ld_point_repeat
*      ,Member,n_member,est_ddisp_point,ak_nonlinear)
implicit real*8(A-H,O-Z)
include "..¥..¥sf3st¥submain.h"

```

```

      record / member_s      / Member
      dimension Member(n_member)
      real*8 ld_point_repeat(*), est_ddisp_point(*)
      dimension u(12), ak_nonlinear(12, 12, *)

```

```

C
c      ld_point_repeat(*)      : real*8      線形右辺項ベクトル
c      Member                  : structure
c      n_member                : integer      部材数
c      est_ddisp_point         : real*8
c      ak_nonlinear(*)         : real*8      接線剛性行列 (釣合系)

```

```

      do i=n_member1, n_member2
      ipp=0
      do j=1, 12
      irest = Member(i).irest(j)
      if(irest.gt.0) then
      u(j)=est_ddisp_point(irest)
      else
      u(j)=0.0
      endif
      end do
      do j=1, 12
      irest = Member(i).irest(j)
      if(irest.gt.0) then
      sum=0.0
      do k=1, 12
      sum=sum+ak_nonlinear(j, k, i)*u(k)
      end do
      ld_point_repeat(irest)=ld_point_repeat(irest) - sum
      end if
      end do

      end do
      return
      end

```

8. Cal_stress_pa

```

C
C      ● SUBROUTINE /Cal_stress_pa Parallel version (チェック OK)

```

```

C      ● 部材内応力の計算(ok)

```

```

      subroutine Cal_stress_pa (n_member1, n_member2,
*                               Member, n_member, Model_type, Element,
*                               past_disp_point, past_vel_point,
*                               est_ddisp_point, rot_memb,
*                               E_model6_real, ak_nonlinear)
      implicit real*8 (A-H, O-Z)
      include "..¥..¥sf3st¥submain.h"

```

```

      include "..¥..¥sf3st¥submain.h"
      record / member_s      / Member
      record / element_s     / Element
      record / n_model_s     / Model_type
      record / e_model6_real_s / E_model6_real
      dimension Member(*), Element(*), E_model6_real(*)
      dimension rot_memb(3, 3, 2, *), ak_nonlinear(12, 12, *)
      dimension past_disp_point(*), past_vel_point(*), est_ddisp_point(*),
*                               v(12), vv(12), vp(12), vpp(12), ak(12, 12)

```

```

C
c      モデル番号
c      Model_No. 1 = 1          ! 幾何学非線形型有限要素弾性モデル
c      Model_No. 2 = 2          ! 3次元せん断弾塑性モデル
c      Model_No. 3 = 3          ! 3次元軸力弾塑性モデル
c      Model_No. 4 = 4          ! 3次元ケーブル弾塑性モデル
c      Model_No. 5 = 5          ! 3次元免振モデル (MSS モデル)
c      Model_No. 6 = 6          ! 3次元制震 Maxwell モデル
c      Model_No. 7 = 7          ! 3次元弾塑性バネモデル(*)
c
c      Model_No. 11 = 11         ! 両端ファイバーモデル
c      Model_No. 12 = 12         ! 両端、中央ファイバーモデル
c
c      Model_No. 21 = 21         ! 両端 MS モデル
c      Model_No. 22 = 22         ! 両端、中央 MS モデル
c
c      Model_No. 31 = 31         ! 幾何学非線形+弾塑性型有限要素モデル
c
c      Model_No. 51 = 51         ! 3次元プレテンション動作モデル
c
c      Model_No. 1001 = 1001     ! DLL 有限要素弾塑性モデル

```

要素数

```

      structure / element_s/
c      integer element_type ! 要素タイプ
c      integer n_element    ! 非線形要素番号
c      real*8 E              ! ヤング係数
c      real*8 G              ! せん断係数
c      real*8 A              ! 断面積
c      real*8 Rlx            ! ねじり剛性
c      real*8 Rly            ! y 軸断面二次モーメント
c      real*8 Rlz            ! z 軸断面二次モーメント
c      real*8 AM             ! 単位長さ当たりの質量
c      integer nm_damp       ! 部材減衰の有無
c      end structure
c      record /element_s/ Element
c      ALLOCATABLE ::Element(:)
c      ALLOCATE (Element(n_element))

```

```

C      部材数

```


付 分散並列型動的ソルバーの各サブルーチン

```

c      structure / member_s/
c      integer nm_element      ! 要素番号
c      integer element_type    ! 要素タイプ
c      integer n_element_type  ! 要素タイプ別番号
c      integer nm_dll_element  ! DLL を用いた要素か (0 : システム内要素、1 : DLL 要素)
c      integer nm_point(2)     ! 節点番号
c      integer irest(12)       ! 部材両端の自由度番号表
c      integer nm_analysis     ! 部材解析種別
c      integer nm_group        ! 部材グループ
c      integer nm_local_coord(2) ! 局所座標系の有無とその回転行列の番号
c      integer nm_damp         ! 部材減衰の有無とその減衰行列の番号
c      real*8 alength          ! 長さ
c      real*8 rot_x            ! 部材主軸の回転角度 (度)
c      real*8 force(12)        ! 部材両端の部材端力
c      end structure
c      record / member_s / Member
c      ALLOCATABLE :: Member (:)
c      ALLOCATE (Member (n_member))
c
c      モデルパラメータ
c      structure / n_model_s/
c      integer n_e_models      ! 要素モデルの最大数
c      integer no_e_model(20)  ! 要素モデルの番号
c      integer n_div_model(20) ! 要素モデルの分割数
c      integer nm_div_model(20) ! 要素モデル内のサブ要素の分割数
c      integer n_e_model(20)   ! 要素モデルの数
c      integer n_m_model(20)   ! 部材モデルの数
c      integer n_damp          ! 部材減衰ありか
c      end structure
c      record / n_model_s / Model_type
c
c      ak_nonlinear      real*8 接線剛性行列
c      Member            structure
c      n_member          integer 部材数
c      Model_type        structure
c      Element           structure
c      past_disp_point(*) real*8 増分前の変位
c      rot_memb          real*8 回転行列
c
c      work1_elememt     real*8 DLL 用ワークエリア
c      work2_elememt     real*8 DLL 用ワークエリア
c      work1_member      real*8 DLL 用ワークエリア
c      work2_member      real*8 DLL 用ワークエリア
c
c      write(76,*) ' 部材応力: ', n_member
c      do i=n_member1, n_member2
c
c      ★部材両端の変位取得
c      do j=1, 12
c      ires=Member(i).irest(j)
c      if(ires.gt.0) then

```

Manual of distributed parallel processing system for dynamic analysis

```

vp(j)=past_disp_point(ires)
v(j)= est_ddisp_point(ires)
else
v(j)=0.
vp(j)=0.
endif
enddo

c      ★変位を釣合系から部材座標系に変換

call RotateL_v(1,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vv)
call RotateL_v(1,vp,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vvp)

c      ★要素及びモデルのセット

mem = i
iet = Member(i).element_type
iet=(iet-1)/10
ie = Member(i).nm_element
im = Element(ie).n_element
imm = Member(i).n_element_type
ien= Member(i).n_model_type
if(Member(i).nm_dll_element.ne. 0) goto 9999 ! DLL 要素
if(iett.eq.0)then
goto(11,12,13,14,15,16,17,18,19,20), iet
11 continue

c      ★Model_No.1 通常の有限要素弾塑性モデル

call Cal_stress_M1(Member(i),Element(ie),
*      ak_nonlinear(1,1,i),v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
goto 100
12 continue

c      ★Model_No.2 3次元せん断弾塑性モデル

call Cal_stress_M2(Member(i),Element(ie),vv)
goto 100
13 continue

c      ★Model_No.3 3次元軸力弾塑性モデル

call Cal_stress_M3(Member(i),Element(ie),
*      ak_nonlinear(1,1,i),v,vv,vvp,
*      rot_memb(1,1,1,i),rot_memb(1,1,2,i))
c      call Cal_stress_M3x(Member(i),Element(ie),
c      *      ak_nonlinear(1,1,i),v,vv,vvp)
goto 100
14 continue

c      ★Model_No.4 3次元ケーブル弾塑性モデル

call Cal_stress_M4(Member(i),Element(ie),vv)
goto 100
15 continue

c      ★Model_No.5 3次元免振モデル

call Cal_stress_M5(Member(i),Element(ie),

```

```

      *      ak_nonlinear(1,1,i),v,rot_memb(1,1,1,i),rot_memb(1,1,2,i) )
      goto 100
16 continue

c-----★Model_No.6 3次元制震 Maxwell モデル
(ok)
      do j=1,12
      ires=Member(i).irest(j)
      if(ires.gt.0) then
      v(j)=past_vel_point(ires)
      else
      v(j)=0.
      endif
      enddo
      call RotateL_v(1,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vv)

      ii=Element(ie).nm_type
c      write(76,'(a,2i4,f10.4)') 'maxwell ',i,ien,E_model6_real(ien).gumma
      call Stress_maxwelldamp(vpp,vv,E_model6_real(ien),
      *      Element(ie),ii,Member(i),Model_type.n_m_filter)
      goto 100
17 continue

c-----★Model_No.7 3次元プレテンション動作
モデル
c      call Cal_stress_M7(Member(i),Element(ie),
c      *      E_model1_int(im),E_model1_real(im),
c      *      M_model1_int(imm),M_model1_int(imm),
c      *      vv,ak )
      goto 100
18 continue

c-----★Model_No.8

      goto 100
19 continue

c-----★Model_No.9

      goto 100
20 continue

c-----★Model_No.10

      goto 100

      elseif(iett.eq.1)then
      goto(111,112,113,114,115,116,117,118,119,120),iet-10
111 continue

c-----★Model_No.11 両端ファイバーモデル

      call Cal_stress_M11( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      goto 100

```

```

112 continue

c-----★Model_No.12 両端、中央ファイバーモデル

      call Cal_stress_M12( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      do j=1,6
      k=j+12
      Member(i).stress(k)=(Member(i).stress(j)+
      *      Member(i).stress(j+6))*0.5
      enddo
      goto 100
113 continue

c-----★Model_No.13 両端 MS モデル

      call Cal_stress_M21( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      goto 100
114 continue

c-----★Model_No.14 両端、中央 MS モデル

      call Cal_stress_M22( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      do j=1,6
      k=j+12
      Member(i).stress(k)=(Member(i).stress(j)+
      *      Member(i).stress(j+6))*0.5
      enddo
      goto 100
115 continue

c-----★Model_No.15 幾何学非線形+弾塑性型有限要素モ
デル

      call Cal_stress_M15( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      goto 100
116 continue

c-----★Model_No.16

      call Cal_stress_M31( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      goto 100
117 continue

c-----★Model_No.17

      call Cal_stress_M32( Model_type ,Member(i),Element(ie),
      *      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
      do j=1,6
      k=j+12
      Member(i).stress(k)=(Member(i).stress(j)+
      *      Member(i).stress(j+6))*0.5
      enddo
      goto 100
118 continue

c-----★Model_No.18

```

```

      call Cal_stress_M13( Model_type ,Member(i),Element(ie),
*      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))
c      do j=1,6
c      k=j+12
c      Member(i).stress(k)=(Member(i).stress(j)+
c      *      Member(i).stress(j+6))*0.5
c      enddo
c      goto 100
119 continue
c-----★Model_No. 19
      call Cal_stress_M33( Model_type ,Member(i),Element(ie),
*      ak_nonlinear(1,1,i) ,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i))

      goto 100
120 continue
c-----★Model_No. 20

      goto 100

      endif
      goto 100

9999 continue
c-----★Model_No. DLL
c      call Cal_stress_dll(mem,Member(i),Element(ie),
c      *      work1_element,work2_element,work1_member,work2_member,
c      *      vv,ak)

c      write(76,'(i4,12e10.3)') i,(Member(i).stress(j),j=1,12)

100 continue
end do
return
end

```

9. Check_stress_pa

```

C
C ● SUBROUTINE /Check_stress Parallel Version (チェック OK)
C
C ● 部材の塑性状態をチェックする(ok)
C
subroutine Check_stress_pa(n_member1,n_member2,Control,N_analysis,
*      ak_nonlinear,Member,n_member,
*      Model_type,Element,past_disp_point,disp_point,rot_memb,
*      E_model6_real,E_model7_real,E_model_fiber,M_model_fiber,
*      E_model11, M_model11,
*      E_model12, M_model12,
*      E_model13, M_model13,
*      E_model15, M_model15,

```

```

*      E_model21, M_model21,
*      E_model22, M_model22,
*      E_model31, M_model31,
*      E_model32, M_model32,
*      E_model33, M_model33,
*      MSS_work,
*      Bilinear_work,Trilinear_work,Concrete_work,R0_work,
*      work1_element,work2_element, work1_member, work2_member )
implicit real*8(A-H,O-Z)
include "..%.%sf3st%submain.h"
include "..%.%sf3st%submainx.h"
record /control_s / Control
record / member_s / Member
record / element_s / Element
record / n_model_s / Model_type
record / Bilinear_work_s / Bilinear_work
record / Trilinear_work_s / Trilinear_work
record / Concrete_work_s / Concrete_work
c-----★Model_No. 1 通常の有限要素弾塑性モデル
ル
c
c-----★Model_No. 2 3次元せん断弾塑性モデル
      record / R0_work_s / R0_work
c-----★Model_No. 3 3次元軸力弾塑性モデル
c      record /element3_s / Element
c      record /member3_s / Member
c-----★Model_No. 4 3次元ケーブル弾塑性モデル
ル
c      record /element4_s / Element
c      record /member4_s / Member
c-----★Model_No. 5 3次元免振モデル
c      record /element5_s / Element
c      record /member5_s / Member
c      record / MSS_work_s / MSS_work
c-----★Model_No. 6 3次元制震 Maxwell モデル
c      record /element6_s / Element
c      record / E_model6_real_s / E_model6_real
c-----★Model_No. 7 3次元パネモデル
c      record /element7_s / Element
c      record /member7_s / Member
c      record / E_model7_real_s / E_model7_real
c-----★Model_No. 11 両端ファイバーモデル
      record / E_model11_s / E_model11
      record / M_model11_s / M_model11
c-----★Model_No. 12 両端、中央ファイバーモデル
ル
      record / E_model12_s / E_model12
      record / M_model12_s / M_model12
c-----★Model_No. 13 両端ピン、中央ファイバー
モデル

```

付 分散並列型動的ソルバーの各サブルーチン

```

c      record / E_model13_s / E_model13
c      record / M_model13_s / M_model13
c      ★Model_No. 15 両端ファイバーFEM モデル
c      record / E_model15_s / E_model15
c      record / M_model15_s / M_model15
c      ★Model_No. 21 両端 MS モデル
c      record / E_model21_s / E_model21
c      record / M_model21_s / M_model21
c      ★Model_No. 22 両端、中央 MS モデル
c      record / E_model22_s / E_model22
c      record / M_model22_s / M_model22
c      ★Model_No. 31 両端アナロジーモデル
c      record / E_model31_s / E_model31
c      record / M_model31_s / M_model31
c      ★Model_No. 32 両端、中央アナロジーモデル
c      record / E_model32_s / E_model32
c      record / M_model32_s / M_model32
c      ★Model_No. 33 両端、中央アナロジーモデル
c      record / E_model33_s / E_model33
c      record / M_model33_s / M_model33
c      ★Model_No. 51 3次元プレテンション動作
c      モデル
c      record / E_model51_s / E_model51
c      record / M_model51_s / M_model51
c      ★ファイバーモデル用エリア
c      record / E_model_fiber_s / E_model_fiber
c      record / M_model_fiber_s / M_model_fiber
c
c      dimension E_model_fiber(*), M_model_fiber(*)
c      dimension Member(*), Element(*), E_model6_real(*)
c      dimension MSS_work(*), R0_work(*)
c      dimension E_model11(*), M_model11(*)
c      dimension E_model12(*), M_model12(*)
c      dimension E_model13(*), M_model13(*)
c      dimension E_model15(*), M_model15(*)
c      dimension E_model21(*), M_model21(*)
c      dimension E_model22(*), M_model22(*)
c      dimension E_model31(*), M_model31(*)
c      dimension E_model32(*), M_model32(*)
c      dimension E_model33(*), M_model33(*)
c      dimension ak_nonlinear(12, 12, *), rot_memb(3, 3, 2, *)
c      dimension work1_element(*), work2_element(*),
c      *      work1_member(*), work2_member(*)
c      dimension past_disp_point(*), disp_point(*)
c      dimension vv(12), vp(12), vpp(12), v(12), ak(12, 12)
c      dimension f(12), ff(12)
c
c      Model_No. 1 = 1      ! 通常の有限要素弾塑性モデル

```

Manual of distributed parallel processing system for dynamic analysis

```

c      Model_No. 2 = 2      ! 3次元せん断弾塑性モデル
c      Model_No. 3 = 3      ! 3次元軸力弾塑性モデル
c      Model_No. 4 = 4      ! 3次元ケーブル弾塑性モデル
c      Model_No. 5 = 5      ! 3次元免振モデル
c      Model_No. 6 = 6      ! 3次元制震 Maxwell モデル
c      Model_No. 7 = 7      ! 3次元プレテンション動作モデル
c
c      要素数
c      structure / element_s/
c      integer element_type ! 要素タイプ
c      integer n_element    ! 非線形要素番号
c      real*8 E              ! ヤング係数
c      real*8 G              ! せん断係数
c      real*8 A              ! 断面積
c      real*8 Rlx            ! ねじり剛性
c      real*8 Rly            ! y 軸断面二次モーメント
c      real*8 Rlz            ! z 軸断面二次モーメント
c      real*8 AM             ! 単位長さ当たりの質量
c      integer nm_damp       ! 部材減衰の有無
c      end structure
c      record /element_s/ Element
c      ALLOCATABLE ::Element(:)
c      ALLOCATE (Element(n_element))
c
c      部材数
c      structure / member_s/
c      integer nm_element    ! 要素番号
c      integer element_type  ! 要素タイプ
c      integer n_element_type ! 要素タイプ別番号
c      integer nm_dll_element ! DLL を用いた要素か (0 ; システム内要素、1 : DLL 要素)
c      integer nm_point(2)   ! 節点番号
c      integer irest(12)     ! 部材両端の自由度番号表
c      integer nm_analysis   ! 部材解析種別
c      integer nm_group      ! 部材グループ
c      integer nm_local_coord(2) ! 局所座標系の有無とその回転行列の番号
c      integer nm_damp       ! 部材減衰の有無とその減衰行列の番号
c      real*4 alength        ! 長さ
c      real*4 rot_x          ! 部材主軸の回転角度 (度)
c      real*8 force(12)      ! 部材両端の部材端力
c      end structure
c      record / member_s / Member
c      ALLOCATABLE :: Member (:)
c      ALLOCATE (Member (n_member))
c
c      モデルパラメータ
c      structure / n_model_s/
c      integer n_e_models    ! 要素モデルの最大数
c      integer no_e_model(20) ! 要素モデルの番号
c      integer n_div_model(20) ! 要素モデルの分割数
c      integer nm_div_model(20) ! 要素モデル内のサブ要素の分割数

```

```

c      integer      n_e_model(20)      ! 要素モデルの数
c      integer      n_m_model(20)      ! 部材モデルの数
c      integer      n_damp              ! 部材減衰ありか
c      end structure
c      record / n_model_s / Model_type
c
c
c      ak_nonlinear      real*8      接線剛性行列
c      Member            structure
c      n_member          integer      部材数
c      Model_type        structure
c      Element           structure
c      past_disp_point(*) real*8      増分前の変位
c      rot_memb          real*8      回転行列
c      E_model1_int      integer      要素モデル 1
c      E_model1_real     real*8      要素モデル 1
c      M_model1_int      integer      部材モデル 1
c      M_model1_real     real*8      部材モデル 1
c      E_model2_int      integer      要素モデル 2
c      E_model2_real     real*8      要素モデル 2
c      M_model2_int      integer      部材モデル 2
c      M_model2_real     real*8      部材モデル 2
c      E_model3_int      integer      要素モデル 3
c      E_model3_real     real*8      要素モデル 3
c      M_model3_int      integer      部材モデル 3
c      M_model3_real     real*8      部材モデル 3
c      E_model4_int      integer      要素モデル 4
c      E_model4_real     real*8      要素モデル 4
c      M_model4_int      integer      部材モデル 4
c      M_model4_real     real*8      部材モデル 4
c      E_model5_int      integer      要素モデル 5
c      E_model5_real     real*8      要素モデル 5
c      M_model5_int      integer      部材モデル 5
c      M_model5_real     real*8      部材モデル 5
c      E_model6_int      integer      要素モデル 6
c      E_model6_real     real*8      要素モデル 6
c      M_model6_int      integer      部材モデル 6
c      M_model6_real     real*8      部材モデル 6
c      work1_element     real*8      DLL 用ワークエリア
c      work2_element     real*8      DLL 用ワークエリア
c      work1_member      real*8      DLL 用ワークエリア
c      work2_member      real*8      DLL 用ワークエリア
c

```

C ————— ★部材両端の変位取得

C ————— ★部材両端の変位取得

12 continue

```

do j=1,3
  f(j)=-Member(i).stress(j)
  f(j+6)=-f(j)
enddo
call Rotatel_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
do j=1,12
  Member(i).force(j)=ff(j)
enddo
goto 100
13 continue
c-----★Model_No.3 3次元軸力弾塑性モデル
c
  iee = Member(i).n_model_type
  call Cal_check_stiff_M3(Member(i),Element(ie),
*    vv,vpp,N_analysis)
  do j=1,12
    f(j)=0.
  enddo
  do j=1,3
    f(j)=-Member(i).stress(j)
    f(j+6)=Member(i).stress(j+3)
  enddo
  call Rotatel_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
  do j=1,12
    Member(i).force(j)=ff(j)
  enddo
  goto 100
14 continue
c-----★Model_No.4 3次元ケーブル弾塑性モデル
c
  call Cal_check_stiff_M4(Member(i),Element(ie),
*    vv,vpp,N_analysis)

  f(1)=-Member(i).stress(1)
  f(7)=-f(1)
  call Rotatel_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
  do j=1,12
    Member(i).force(j)=ff(j)
  enddo
  goto 100
15 continue

c-----★Model_No.5 3次元免振モデル
c
  if(N_analysis.ge.9) then
    iee=Member(i).n_model_type
c
    write(76,'(a,i4)') '***** mss iee check',iee,i
    call Cal_check_stiff_M5(Member(i),Element(ie),MSS_work(iee),
*    vv,vpp,Model_type.n_spring,Model_type.cosin(1,1) )
  endif
  do j=1,3

```

```

  f(j)=-Member(i).stress(j)
  f(j+6)=-f(j)
enddo
call Rotatel_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
do j=1,12
  Member(i).force(j)=ff(j)
enddo
goto 100
16 continue
c-----★Model_No.6 3次元制震Maxwellモデル
(ok)
c
  call Check_maxwelldamp(E_model6_real(ien),Element(ie))

  goto 100
17 continue
c-----★Model_No.7 3次元プレテンション動作モデル
c
  call Cal_check_stiff_M7(Member(i),Element(ie),
c
*    E_model1_int(im),E_model1_real(im),
c
*    M_model1_int(imm),M_model1_int(imm),
c
*    vv,ak )
c
  do j=1,12
    f(j)=0.
  enddo
  do j=1,3
    f(j)=-Member(i).stress(j)
    f(j+6)=-f(j)
  enddo
  call Rotatel_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
  do j=1,12
    Member(i).force(j)=ff(j)
  enddo
  goto 100
18 continue
c-----★Model_No.8

  goto 100
19 continue
c-----★Model_No.9

  goto 100
20 continue
c-----★Model_No.10

  goto 100

elseif(iett.eq.1) then
c
  write(76,'(a,i3,12e12.5)') 'iet',iet
  goto(111,112,113,114,115,116,117,118,119,120),iet-10

```

付 分散並列型動的ソルバーの各サブルーチン

```

111 continue
c-----★Model_No. 11 両端ファイバーモデル
    call Cal_check_stiff_M11(Control,N_analysis,
    *   mem,Model_type,Member(i),Element(ie),
    *   E_model11, E_model_fiber,
    *   M_model11, M_model_fiber,
    *   Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)
c-----★部材の両端節点力を釣合系に変換
c    write(76,'(a,i3,12e12.5)') 'mem',i,(f(j),j=1,6)
c    write(76,'(a,i3,12e12.5)') 'mem',i,(f(j),j=7,12)

    call RotateL_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
    do j=1,12
    Member(i).force(j)=Member(i).force(j)+ff(j)
    enddo
c    write(76,'(a,i3,12e12.5)') 'mem',i,(Member(i).force(j),j=1,6)
c    write(76,'(a,i3,12e12.5)') 'mem',i,(Member(i).force(j),j=7,12)
    goto 100
112 continue
c-----★Model_No. 12 両端、中央ファイバーモデル
    call Cal_check_stiff_M12(Control,N_analysis,
    *   mem,Model_type,Member(i),Element(ie),
    *   E_model12, E_model_fiber,
    *   M_model12, M_model_fiber,
    *   Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)

    vv11=vv(8)-vv(2)  ! 非線形剛性計算のための変位セット
    vv22=vv(9)-vv(3)
c    write(76,'(a,10f12.4)') 'an ',Member(i).stress(7),vv11,vv22
c-----★部材の両端節点力を釣合系に変換
c    write(76,'(a,i3,12e12.5)') 'mem',i,(f(j),j=1,6)
c    write(76,'(a,i3,12e12.5)') 'mem',i,(f(j),j=7,12)
c-----★非線形軸力の計算
c    if(N_analysis.ne.9.and.N_analysis.ne.7)
c    *   call nonlinear_stress_F(Element(ie),Member(i),vv,f)

    call RotateL_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
    do j=1,12
    Member(i).force(j)=Member(i).force(j)+ff(j)
    enddo
c    write(76,'(a,i3,12e12.5)') 'mem',i,(Member(i).force(j),j=1,6)
c    write(76,'(a,i3,12e12.5)') 'mem',i,(Member(i).force(j),j=7,12)

    goto 100
113 continue
c-----★Model_No. 13 両端 MS モデル
    call Cal_check_stiff_M21(N_analysis,
    *   mem,Model_type,Member(i),Element(ie),
    *   E_model21, E_model_fiber,
    *   M_model21, M_model_fiber,

```

Manual of distributed parallel processing system for dynamic analysis

```

    *   Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)
c-----★部材の両端節点力を釣合系に変換
    call RotateL_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
    do j=1,12
    Member(i).force(j)=Member(i).force(j)+ff(j)
    enddo

    goto 100
114 continue
c-----★Model_No. 14 両端、中央 MS モデル

    call Cal_check_stiff_M22(N_analysis,
    *   mem,Model_type,Member(i),Element(ie),
    *   E_model22, E_model_fiber,
    *   M_model22, M_model_fiber,
    *   Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)
c-----★部材の両端節点力を釣合系に変換
    call RotateL_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
    do j=1,12
    Member(i).force(j)=Member(i).force(j)+ff(j)
    enddo
    goto 100
115 continue
c-----★Model_No. 15 幾何学非線形+弾塑性型有限要素モデル
ル
    call Cal_check_stiff_M15(Control,N_analysis,
    *   mem,Model_type,Member(i),Element(ie),
    *   E_model15, E_model_fiber,
    *   M_model15, M_model_fiber,
    *   Bilinear_work,Trilinear_work,Concrete_work,vv,vpp,f)
c-----★部材の両端節点力を釣合系に変換
    do j=1,6
    f(j)=-Member(i).stress(j)
    enddo
    do j=7,12
    f(j)=Member(i).stress(j)
    enddo
    call RotateL_v(2,f,rot_memb(1,1,1,i),rot_memb(1,1,2,i),ff)
    do j=1,12
    Member(i).force(j)= ff(j)
    enddo
c    write(76,'(a,i3,12e12.5)') 'mem',i,(Member(i).force(j),j=1,6)
c    write(76,'(a,i3,12e12.5)') 'mem',i,(Member(i).force(j),j=7,12)

    goto 100
116 continue
c-----★Model_No. 16 両端アナロジーモデル
    call Cal_check_stiff_M31(N_analysis,
    *   mem,Model_type,Member(i),Element(ie),
    *   E_model31, E_model_fiber,

```

```

*      M_model131, M_model_fiber,
*      vv, vpp, f)
c-----★部材の両端節点力を釣合系に変換
call RotateL_v(2, f, rot_memb(1, 1, 1, i), rot_memb(1, 1, 2, i), ff)
do j=1, 12
  Member(i).force(j)=Member(i).force(j)+ff(j)
enddo
goto 100
goto 100
117 continue

c-----★Model_No. 17 両端、中央アナロジーモデル
call Cal_check_stiff_M32(N_analysis,
*      mem, Model_type, Member(i), Element(ie),
*      E_model132, E_model_fiber,
*      M_model132, M_model_fiber,
*      vv, vpp, f)
c-----★部材の両端節点力を釣合系に変換
call RotateL_v(2, f, rot_memb(1, 1, 1, i), rot_memb(1, 1, 2, i), ff)
do j=1, 12
  Member(i).force(j)=Member(i).force(j)+ff(j)
enddo
118 continue

c-----★Model_No. 18
call Cal_check_stiff_M13(Control, N_analysis,
*      mem, Model_type, Member(i), Element(ie),
*      E_model13, E_model_fiber,
*      M_model13, M_model_fiber,
*      Bilinear_work, Trilinear_work, Concrete_work, vv, vpp, f)
call RotateL_v(2, f, rot_memb(1, 1, 1, i), rot_memb(1, 1, 2, i), ff)
do j=1, 12
  Member(i).force(j)=Member(i).force(j)+ff(j)
enddo
goto 100
119 continue

c-----★Model_No. 19 両端、中央アナロジーモデル
call Cal_check_stiff_M33(N_analysis,
*      mem, Model_type, Member(i), Element(ie),
*      E_model133, E_model_fiber,
*      M_model133, M_model_fiber,
*      vv, vpp, f)
c-----★部材の両端節点力を釣合系に変換
call RotateL_v(2, f, rot_memb(1, 1, 1, i), rot_memb(1, 1, 2, i), ff)
do j=1, 12
  Member(i).force(j)=Member(i).force(j)+ff(j)
enddo
goto 100
120 continue

```

```

c-----★Model_No. 20
goto 100

endif
goto 100
9999 continue

c-----★Model_No. DLL
c      call Cal_check_stiff_dll(mem, Member(i), Element(ie),
c      *      work1_element, work2_element, work1_member, work2_member,
c      *      vv, ak)

c-----★部材の接線剛性を釣合系に変換
100 continue

end do
return
end

C
C ● SUBROUTINE /Check_Maxwell_stress (チェック OK)
C
C ● Maxwell model の応力計算
C
subroutine Check_Maxwell_stress_pa(n_member1, n_member2, Member,
*      Element, E_model6_real, Newmark_P)
implicit real*8(A-H, O-Z)
include "..¥..¥sf3st¥submain.h"
record / member_s / Member
record / element_s / Element
record / E_model6_real_s / E_model6_real
record / newmark_s / Newmark_P
dimension Member(*), Element(*), E_model6_real(*)

c-----★要素及びモデルのセット
do i=n_member1, n_member2
c-----★Model_No. 6 3次元制震 Maxwell モデル
(ok)
call Check_maxwelldamp(E_model6_real(ien), Element(ie), Newmark_P)
endif
end do
return
end

```


1 0. Get_max_stress_pa

```

C
C  ● SUBROUTINE /Get_max_stress_pa
C
C  ● 最大応力を求める (ok)
C
subroutine Get_max_stress_pa(n_member1,n_member2,
*                               Member,Max_stress)
implicit real*8 (A-H,O-Z)
include "..¥..¥sf3st¥submain.h"
record / member_s / Member
record / max_stress_s / Max_stress
dimension Member(*),Max_stress(*)

C
c Max_stress      :structure 最大値
c n_member        :integer   節点数
C

do i=n_member1,n_member2
do j=1,18
aa = Member(i).stress(j)
if(Max_stress(i).stress_max_member(j).lt.aa)
*      Max_stress(i).stress_max_member(j)=aa
if(Max_stress(i).stress_min_member(j).gt.aa)
*      Max_stress(i).stress_min_member(j)=aa
end do
end do
return
end

```

1 1. Get_nonlinear_stiff_pa

```

C
C  ● SUBROUTINE /Get_nonlinear_stiff_pa Parallel Version
C
C  ● 接線剛性行列の計算 (ok)
C
subroutine Get_nonlinear_stiff(N_analysis,
C *      ak_nonlinear,Member,n_member,
C *      Model_type,Element,past_disp_point,disp_point,rot_memb,
C *      E_model6_real,E_model7_real,E_model_fiber,M_model_fiber,
C *      E_model11, M_model11,
C *      E_model12, M_model12,
C *      E_model15, M_model15,
C *      E_model21, M_model21,
C *      E_model22, M_model22,
C *      E_model31, M_model31,
C *      E_model32, M_model32,
C *      MSS_work,
C *      work1_element,work2_element, work1_member, work2_member)
subroutine Get_nonlinear_stiff_pa(n_member1,n_member2,N_analysis,

```

```

*      ak_nonlinear,Member,n_member,
*      Model_type,Element,past_disp_point,disp_point,rot_memb,
*      E_model6_real,E_model7_real,E_model_fiber,M_model_fiber,
*      E_model11, M_model11,
*      E_model12, M_model12,
*      E_model13, M_model13,
*      E_model15, M_model15,
*      E_model21, M_model21,
*      E_model22, M_model22,
*      E_model31, M_model31,
*      E_model32, M_model32,
*      E_model33, M_model33,
*      MSS_work,
*      work1_element,work2_element, work1_member, work2_member)
implicit real*8 (A-H,O-Z)
include "..¥..¥sf3st¥submain.h"
include "..¥..¥sf3st¥submainx.h"
record / member_s / Member
record / element_s / Element
record / n_model_s / Model_type

```

```

C-----★Model_No.1 通常の有限要素弾塑性モデル
ル
C
C-----★Model_No.2 3次元せん断弾塑性モデル
c      record /element2_s / Element
c      record /member2_s / Member
C-----★Model_No.3 3次元軸力弾塑性モデル
c      record /element3_s / Element
c      record /member3_s / Member
C-----★Model_No.4 3次元ケーブル弾塑性モデル
ル
c      record /element4_s / Element
c      record /member4_s / Member
C-----★Model_No.5 3次元免振モデル
c      record /element5_s / Element
c      record /member5_s / Member
c      record / MSS_work_s / MSS_work
C-----★Model_No.6 3次元制震 Maxwell モデル
c      record /element6_s / Element
c      record / E_model6_real_s / E_model6_real
C-----★Model_No.7 3次元バネモデル
c      record /element7_s / Element
c      record /member7_s / Member
c      record / E_model7_real_s / E_model7_real
C-----★Model_No.11 両端ファイバーモデル
c      record / E_model11_s / E_model11
c      record / M_model11_s / M_model11
C-----★Model_No.12 両端、中央ファイバーモデル
ル

```

付 分散並列型動的ソルバーの各サブルーチン

```

      record / E_model12_s / E_model12
      record / M_model12_s / M_model12
c-----★Model_No. 13 両端ピン、中央ファイバー
モデル
      record / E_model13_s / E_model13
      record / M_model13_s / M_model13
c-----★Model_No. 15 両端ファイバーFEM モデル
      record / E_model15_s / E_model15
      record / M_model15_s / M_model15
c-----★Model_No. 21 両端 MS モデル
      record / E_model21_s / E_model21
      record / M_model21_s / M_model21
c-----★Model_No. 22 両端、中央 MS モデル
      record / E_model22_s / E_model22
      record / M_model22_s / M_model22
c-----★Model_No. 31 両端アナロジーモデル
      record / E_model31_s / E_model31
      record / M_model31_s / M_model31
c-----★Model_No. 32 両端、中央アナロジーモデル
      record / E_model32_s / E_model32
      record / M_model32_s / M_model32
c-----★Model_No. 33 両端、中央アナロジーモデル
      record / E_model33_s / E_model33
      record / M_model33_s / M_model33
c-----★Model_No. 51 3 次元プレテンション動作
モデル
c      record / E_model51_s / E_model51
c      record / M_model51_s / M_model51
c-----★ファイバーモデル用エリア
      record / E_model_fiber_s / E_model_fiber
      record / M_model_fiber_s / M_model_fiber
c-----
dimension Member(*), Element(*), MSS_work(*)
dimension ak_nonlinear(12, 12, *), rot_memb(3, 3, 2, *)
dimension work1_element(*), work2_element(*),
*      work1_member(*), work2_member(*)
dimension past_disp_point(*), disp_point(*), v(12), ak(12, 12)
dimension vv(12)
c-----
c 要素数
c      structure / element_s/
c      integer element_type ! 要素タイプ
c      integer n_element ! 非線形要素番号
c      real*8 E ! ヤング係数
c      real*8 G ! せん断係数
c      real*8 A ! 断面積
c      real*8 Rlx ! ねじり剛性
c      real*8 Rly ! y 軸断面二次モーメント

```

Manual of distributed parallel processing system for dynamic analysis

```

c      real*8 Rlz ! z 軸断面二次モーメント
c      real*8 AM ! 単位長さ当たりの質量
c      integer nm_damp ! 部材減衰の有無
c      end structure
c      record /element_s/ Element
c      ALLOCATABLE ::Element(:)
c      ALLOCATE (Element(n_element))
c
c      部材数
c      structure / member_s/
c      integer nm_element ! 要素番号
c      integer element_type ! 要素タイプ
c      integer n_element_type ! 要素タイプ別番号
c      integer nm_dll_element ! DLL を用いた要素か (0 ; システム内要素、1 : DLL 要素)
c      integer nm_point(2) ! 節点番号
c      integer irest(12) ! 部材両端の自由度番号表
c      integer nm_analysis ! 部材解析種別
c      integer nm_group ! 部材グループ
c      integer nm_local_coord(2) ! 局所座標系の有無とその回転行列の番号
c      integer nm_damp ! 部材減衰の有無とその減衰行列の番号
c      real*8 alength ! 長さ
c      real*8 rot_x ! 部材主軸の回転角度 (度)
c      real*8 force(12) ! 部材両端の部材端力
c      end structure
c      record / member_s / Member
c      ALLOCATABLE :: Member (:)
c      ALLOCATE (Member (n_member))
c
c      モデルパラメータ
c      structure / n_model_s/
c      integer n_e_models ! 要素モデルの最大数
c      integer no_e_model(20) ! 要素モデルの番号
c      integer n_div_model(20) ! 要素モデルの分割数
c      integer nm_div_model(20) ! 要素モデル内のサブ要素の分割数
c      integer n_e_model(20) ! 要素モデルの数
c      integer n_m_model(20) ! 部材モデルの数
c      integer n_damp ! 部材減衰ありか
c      end structure
c      record / n_model_s / Model_type
c
c      ak_nonlinear real*8 接線剛性行列
c      Member structure
c      n_member integer 部材数
c      Model_type structure
c      Element structure
c      past_disp_point(*) real*8 増分前の変位
c      rot_memb real*8 回転行列
c      E_model1_int integer 要素モデル 1

```

付 分散並列型動的ソルバーの各サブルーチン

```

c      E_model1_real      real*8      要素モデル 1
c      M_model1_int       integer     部材モデル 1
c      M_model1_real      real*8      部材モデル 1
c      E_model2_int       integer     要素モデル 2
c      E_model2_real      real*8      要素モデル 2
c      M_model2_int       integer     部材モデル 2
c      M_model2_real      real*8      部材モデル 2
c      E_model3_int       integer     要素モデル 3
c      E_model3_real      real*8      要素モデル 3
c      M_model3_int       integer     部材モデル 3
c      M_model3_real      real*8      部材モデル 3
c      E_model4_int       integer     要素モデル 4
c      E_model4_real      real*8      要素モデル 4
c      M_model4_int       integer     部材モデル 4
c      M_model4_real      real*8      部材モデル 4
c      E_model5_int       integer     要素モデル 5
c      E_model5_real      real*8      要素モデル 5
c      M_model5_int       integer     部材モデル 5
c      M_model5_real      real*8      部材モデル 5
c      E_model6_int       integer     要素モデル 6
c      E_model6_real      real*8      要素モデル 6
c      M_model6_int       integer     部材モデル 6
c      M_model6_real      real*8      部材モデル 6
c      work1_elememt      real*8      DLL 用ワークエリア
c      work2_elememt      real*8      DLL 用ワークエリア
c      work1_member       real*8      DLL 用ワークエリア
c      work2_member       real*8      DLL 用ワークエリア
C

```

```

if(N_analysis.eq.7) return      ! 線形解析:7
do 9999 i=n_member1,n_member2
if(Member(i).nm_analysis .eq. -1) goto 9999 ! 各部材の解析種別のチェック (-1:線形解析)
C-----★部材両端の変位取得
do j=1,12
ires=Member(i).irest(j)
if(ires.gt.0) then
v(j)=past_disp_point(ires)
else
v(j)=0.
endif
enddo
C-----★変位を釣合系から部材座標系に変換

call RotateL_v(1,v,rot_memb(1,1,1,i),rot_memb(1,1,2,i),vv)

C-----★要素及びモデルのセット
mem = i
iet = Member(i).element_type
iet=(iet-1)/10
ie = Member(i).nm_element

```

Manual of distributed parallel processing system for dynamic analysis

```

im = Element(ie).n_element
imm = Member(i).n_element_type
ien= Member(i).n_model_type
if(Member(i).nm_dll_element .ne. 0) goto 9998 ! DLL 要素

if(iett.eq.0)then
goto(11,12,13,14,15,16,17,18,19,20), iet
11 continue
C-----★Model_No.1 通常の有限要素弾塑性モデル
ル
if(N_analysis.eq.9) goto 9999 ! 弾塑性解析はなし
Member(i).an_vv(1)=vv(8)-vv(2) ! 非線形剛性計算のための変位セット
Member(i).an_ww(1)=vv(9)-vv(3)
call Cal_nonlin_stiff_M1(Member(i),Element(ie),
* Member(i).stress(7),ak )
if(Member(i).i_rigid_length.ne.0..or.
* Member(i).j_rigid_length.ne.0.)
*call Deal_Rigid_element(ak,
* Member(i).i_rigid_length,Member(i).j_rigid_length)

goto 100
12 continue
C-----★Model_No.2 3次元せん断弾塑性モデル
if(N_analysis.eq.8) goto 9999 ! 幾何学的非線型解析はなし
call Cal_nonlin_stiff_M2(Member(i),Element(ie),
* ak )
goto 100
13 continue
C-----★Model_No.3 3次元軸力弾塑性モデル
c      Member(i).an_vv(1)=vv(8)-vv(2) ! 非線形剛性計算のための変位セット
c      Member(i).an_ww(1)=vv(9)-vv(3)
if(N_analysis.eq.9)then
call Cal_nonlin_stiff_M3(Member(i),Element(ie),ak)
endif
if(N_analysis.eq.8.or.N_analysis.eq.10)then
call Cal_geomet_stiff_Nx(Member(i).stress(1),vv,
* Member(i),Element(ie),ak)
endif
goto 100
14 continue
C-----★Model_No.4 3次元ケーブル弾塑性モデル
ル
if(N_analysis.eq.8) goto 9999 ! 幾何学的非線型解析はなし
call Cal_nonlin_stiff_M4(Member(i),Element(ie),
* ak )
goto 100
15 continue
C-----★Model_No.5 3次元免振モデル
if(N_analysis.eq.8) goto 9999 ! 幾何学的非線型解析はなし
iee=Member(i).n_model_type

```

付 分散並列型動的ソルバーの各サブルーチン

```

      call Cal_nonlin_stiff_M5(Member(i),Element(ie),
      *      Model_type,n_spring,Model_type.cosin(1,1),
      *      MSS_work(iee),ak )
      goto 100
16 continue
c-----★Model_No.6 3次元制震 Maxwell モデル
(ok)
      if(N_analysis.eq.8) goto 9999 ! 幾何学的非線型解析はなし
c      call Cal_link_maxwelldamp(ak) ! 剛性のゼロセット
      goto 9999
17 continue
c-----★Model_No.7 3次元プレテンション動作
モデル
      if(N_analysis.eq.8) goto 9999 ! 幾何学的非線型解析はなし
c      call Cal_nonlin_stiff_M7(Member(i),Element(ie),
c      *      E_model1_int(im),E_model1_real(im),
c      *      M_model1_int(imm),M_model1_int(imm),
c      *      vv,ak )
      goto 100
18 continue
c-----★Model_No.8
      goto 100
19 continue
c-----★Model_No.9
      goto 100
20 continue
c-----★Model_No.10
      goto 100
      elseif(iett.eq.1)then
      goto(111,112,113,114,115,116,117,118,119,120),iet-10
111 continue
c-----★Model_No.11 両端ファイバーモデル
      call Cal_nonlin_stiff_M11(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model11,E_model_fiber,
      *      M_model11,M_model_fiber)
      goto 100
112 continue
c-----★Model_No.12 両端、中央ファイバーモデル
      call Cal_nonlin_stiff_M12(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model12,E_model_fiber,
      *      M_model12,M_model_fiber)
      goto 100

```

Manual of distributed parallel processing system for dynamic analysis

```

113 continue
c-----★Model_No.13 両端 MS モデル
      call Cal_nonlin_stiff_M21(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model21,E_model_fiber,
      *      M_model21,M_model_fiber)
      goto 100
114 continue
c-----★Model_No.14 両端、中央 MS モデル
      call Cal_nonlin_stiff_M22(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model22,E_model_fiber,
      *      M_model22,M_model_fiber)
      goto 100
115 continue
c-----★Model_No.15 幾何学非線形+弾塑性型有
限要素モデル
      call Cal_nonlin_stiff_M15(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model15,E_model_fiber,
      *      M_model15,M_model_fiber)
      goto 100
116 continue
c-----★Model_No.16 両端、中央アナロジーモデル
      call Cal_nonlin_stiff_M31(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model31,E_model_fiber,
      *      M_model31,M_model_fiber)
      goto 100
117 continue
c-----★Model_No.17 両端、中央アナロジーモデル
      call Cal_nonlin_stiff_M32(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model32,E_model_fiber,
      *      M_model32,M_model_fiber)
      goto 100
118 continue
c-----★Model_No.18 両端ピン、中央ファイバー
モデル
      call Cal_nonlin_stiff_M13(N_analysis,
      *      Model_type,Member(i),Element(ie),
      *      ak,ier,E_model13,E_model_fiber,
      *      M_model13,M_model_fiber)
      goto 100
119 continue
c-----★Model_No.19
      call Cal_nonlin_stiff_M33(N_analysis,
      *      Model_type,Member(i),Element(ie),

```

```

*      ak,ier,E_model33, E_model_fiber,
*      M_model33,M_model_fiber)
      goto 100
120 continue
c-----★Model_No. 20

      goto 100

      endif
      goto 100
9998 continue
c-----★Model_No. DLL
c      call Cal_nonlin_stiff_dll(mem,Member(i),Element(ie),
c      *      work1_element,work2_element,work1_member,work2_member,
c      *      vv,ak)

100 continue
c-----★部材の接線剛性を釣合系に変換

      call Rotate_K(ak,rot_memb(1,1,1,i),
*      rot_memb(1,1,2,i),ak_nonlinear(1,1,i))

9999 continue

      return
      end

```

1 2. out_section_check_pa

```

c
c      ● SUBROUTINE /out_section_check_pa
c
c      ● 出力部材の断面応力をチェック
c
c      ● 変更の概要…ファイルへの出力を停止（コメントアウトした）
c      subroutine out_section_check_pa(Member,Element,
*      n_member,No_section,Out_section,
*      ifl,iflz,i_print,n_member1,n_member2)
      implicit real*8(A-H,O-Z)
      include "..¥..¥sf3st¥submain.h"
      record / Member_s / Member
      record / Element_s / Element
      record / Out_section_s / Out_section
      dimension Member(*),Element(*),No_section(*)
      dimension ifl(16),iflz(16)
c
c      ● out_section_s 構造体
c
c      解析パラメータ

```

```

c      structure / out_section_s/
c      integer n_member ! 出力部材数
c      integer n_section ! 出力断面数
c      integer no_member(10) ! 部材番号
c      integer no_section(10) ! 断面数
c      integer no_fiber(30) ! ファイバー数
c      end structure
c      record /out_section_s/ Out_section
c
c
c      if(ifl(6).eq.0) return
      nn=No_section(1)
      if(nn.gt.0) then
        iix=0
        iiy=0
        do i=1,nn
          n=No_section(i+1)
          if(n.ge.n_member1.and.n.le.n_member2)then
            im=Member(n).element_type
            ie= Member(n).nm_element
            if(im.eq.11.or.im.eq.15) then
              iix=iix+1
              iiy=iiy+2
              Out_section.no_member(iix) = n
              Out_section.no_section(iix)= 2
              Out_section.no_fiber(iiy-1)= Element(ie).nm_section(1)
              Out_section.no_fiber(iiy) = Element(ie).nm_section(2)
              Out_section.nm_fiber(iiy-1)= Element(ie).n_section(1)
              Out_section.nm_fiber(iiy) = Element(ie).n_section(2)
            elseif(im.eq.13.or.im.eq.18) then
              iix=iix+1
              iiy=iiy+1
              Out_section.no_member(iix) = n
              Out_section.no_section(iix)= 1
              Out_section.no_fiber(iiy)= Element(ie).nm_section(1)
              Out_section.nm_fiber(iiy)= Element(ie).n_section(1)
            elseif(im.eq.12.or.im.eq.14) then
              iix=iix+1
              iiy=iiy+3
              Out_section.no_member(iix) = n
              Out_section.no_section(iix)= 3
              Out_section.no_fiber(iiy-2)= Element(ie).nm_section(1)
              Out_section.no_fiber(iiy-1)= Element(ie).nm_section(2)
              Out_section.no_fiber(iiy) = Element(ie).nm_section(3)
              Out_section.nm_fiber(iiy-2)= Element(ie).n_section(1)
              Out_section.nm_fiber(iiy-1)= Element(ie).n_section(2)
              Out_section.nm_fiber(iiy) = Element(ie).n_section(3)
            endif
          endif
        enddo
        Out_section.n_member = iix

```

```

    Out_section.n_section = iiy
    else
    Out_section.n_member = 0
    endif
c-----★断面応力ヘッダー出力
    write(iflz(6)) Out_section.n_member
    if(Out_section.n_member.eq.0) return

    write(iflz(6)) (Out_section.no_member(j), j=1, Out_section.n_member)
    write(iflz(6)) (Out_section.no_section(j), j=1, Out_section.n_member)
    write(iflz(6)) Out_section.n_section
    write(iflz(6)) (Out_section.nm_fiber(j), j=1, Out_section.n_section)
    write(iflz(6)) (Out_section.no_fiber(j), j=1, Out_section.n_section)

c-----★断面応力ヘッダー出力終了
c    write(76,'(a)') ' output section '
c    write(76,'(a,i4)') ' No. member: ', Out_section.n_member
c    write(76,'(a,i4)') ' No. section: ', Out_section.n_section
c    if(Out_section.n_member.eq.0) return
c    iiy=0
c    do i=1, Out_section.n_member
c    write(76,'(a,i4)') ' Out_section.no_member', Out_section.no_member(i)
c    write(76,'(a,i4)')
c    *      ' Out_section.no_member', Out_section.no_section(i)
c    do j=1, Out_section.no_section(i)
c    iiy=iiy+1
c    write(76,'(a,i4,i4)')
c    *      ' Out_section.no_fiber', iiy, Out_section.no_fiber(iiy)
c    enddo
c    enddo

    return
end

```